Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 305 (2014) 19–34

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Non-deterministic Semantics in Polynomial Format

Walter Carnielli[1,2] Mariana Matulovic [1,3]

Centre for Logic, Epistemology and the History of Science (CLE) and Department of Philosophy
State University of Campinas- UNICAMP
Campinas, Brazil

Abstract

The method for automatic theorem proving proposed in [6], called *Polynomial Ring Calculus*, is an algebraic proof mechanism based on handling polynomials over finite fields. Although useful in general domains, as in first-order logic, certain non-truth-functional logics and even in modal logics (see [1]), the method is particularly apt for deterministic and non-deterministic many-valued logics, as shown here. The aim of the present paper is to show how the method can be extended to any finite-valued non-deterministic semantics, and also to explore the computational character of the method through the development of a software capable of translating provability in deterministic and non-deterministic finite-valued logical systems into operations on polynomial rings.

Keywords: polynomial proof systems, deterministic and non-deterministic many-valued logics, non-deterministic semantics, satisfiability, complexity.

1 Introduction

Algebraic proof systems based on formal polynomials over algebraically closed fields (the "Polynomial Ring Calculus" - PRC) were introduced in [5] and [6]. Formal polynomials work as a powerful tool for logical derivation in classical and non-classical logics, in particular for propositional (deterministic and non-deterministic) many-valued logics, paraconsistent logics and modal logics.

During the last few years a series of papers and notes concerning the PRC and its applicability have been published. We summarize the main ideas below.

The first sketchy ideas appeared in [5] in 2001, and were subsequently developed in [6]. Polynomial versions for the paraconsistent systems mbC and mCi (particular cases of the Logics of Formal Inconsistency, or LFIs, cf. [9]) were developed in [7],

[1] Thanks to Fapesp, CNPq and CPAI

[2] Email: walter.carnielli@cle.unicamp.br

[3] Email: marianamatulovic@cle.unicamp.br

http://dx.doi.org/10.1016/j.entcs.2014.06.003

where the use of the so-called hidden variables (due to the non-truth-functional character of the paraconsistent semantics) was proposed. A polynomial version for the monadic fragment of first-order logic (FOL) has been given in [8], where it is also shown how any finite function can be expressed by means of polynomials over finite fields (cf. Theorem 3.1, p. 6). The method of polynomials can also be seen as a heuristic device able in some cases to discover new simple logical systems or new properties of logic systems, as shown in [10] and [11]. More recently, in [1] the method of proof by polynomials has been extended to modal logics.

To sum up, the method of polynomial rings is applicable to various logical systems, including classical logic, modal logics, finitely many-valued logics, first-order logic and more surprisingly, as we show here, even to some logics that have logical representation by non-deterministic matrices. There are several reasons why such a new method of proof is worth investigating:

(i) Search for efficiency: the method of polynomial rings can contribute to questions related to computational complexity, as it may shed new light on satisfiability;

(ii) Broadness: as emphasized here, the method of polynomial rings is applicable in various logical systems, and in this way is comparable (in universality) to tableaux;

(iii) Relationships with algebra: the method of polynomial rings recovers an approach to logic via algebra, in the tradition of Boole and Leibniz, that seems to has been forgotten;

(iv) Heuristics: the method of polynomials can be seen as a heuristic method able to discover new logical systems or new properties of logical systems. For examples on how the method can be applied as a heuristic device see [10] and [11].

In this paper we explore the universal and computational character of the method, applying it to deterministic and non-deterministic finite many-valued logic systems. We also show a software, PoLCa, that transforms any finite-valued matrices (deterministic or non-deterministic) into polynomials with coefficients in a Galois field.

2 The Polynomial Ring Calculus- PRC

The method of proof based on polynomial rings (PRC) is an algebraic proof mechanism that works by translating logic formulas of a given language $\mathcal{L}$ into polynomials with coefficients in finite fields, and then performing deductions by accomplishing polynomial operations. Elements of the field represent truth-values, and polynomials may be regarded as the possible truth-values that formulas can take. This makes it possible that truth conditions on formulas can be determined by reducing polynomials through certain operations (the PRC rules). PRC can be regarded as an algebraic semantics, in which the structure of polynomials reflects the structure of truth-value conditions for logic formulas; it can also be seen as a proof method (much as a tableau calculus can be viewed either as a proof-theoretical or as a

model-theoretical device).

We define a (p, m)-Polynomial Rings Calculus ((p, m)-PRC) for a given propositional logic system $\mathcal{L}$, based on the Galois Field, $GF(p^m)$ for p prime and m a natural number other than zero, by the following clauses:

(i) All the (p, m)-PRC terms are variables, and all its formulas are polynomials in $GF(p^m)[X]$;

(ii) Operations in (p, m)-PRC are governed by a set of rules. They are:

(a) Index Rules.
- $p.x \approxeq 0$, where $(p.x)$ means $(x+x+...+x)$, such addition being performed p times.
- $x^i.x^j \approxeq x^k (mod\ q(x))$ in that $q(x)$ is a convenient primitive polynomial that defines $GF(p^m)$, and $k = i + j (mod\ p^m - 1)$.

(b) Ring rules, uniform substitution and Leibniz rules (for equality) [4] .

In this way, the (p, m)-Polynomial Ring Calculus for a given logic $\mathcal{L}$ (written simply as PRC when there is no danger of confusion) basically consists in translating formulas of $\mathcal{L}$ into polynomials with coefficients in a finite field, and performing deductions through operations (governed by the set of rules defined above) on those polynomials. We say that the polynomial rules *prove* a certain sentence α in $\mathcal{L}$ if its translation in reduced form via application of the rules (the polynomial α^* with coefficients in the Galois field $GF(p^m)$) never outputs values outside the set D of distinguished truth-values.

In summary, defining a PRC for a specific logic $\mathcal{L}$ consists in:

- Selecting a suitable finite field, $GF\,(p^m)$, to represent the truth-values, specifying a subset of distinguished (also called designated in the literature) truth-values.
- Defining a translation function from formulas of $\mathcal{L}$ into polynomials with coefficients in $GF\,(p^m)$, namely, $* : Form_{\mathcal{L}} \rightarrow GF\,(p^m)\,[X]$.
- In certain cases, some constraints on translations will have to be added, as in the cases where modal logic are expressed in polynomial format (see [1]) .

The procedure for obtaining a polynomial representation for a (deterministic or non-deterministic) finite-valued logic begins with the construction of truth-tables for each connective in the language that will be translated. From this point on, in order to characterize the polynomials corresponding to formulas, there are two algorithmic options according to which one may proceed: by means of *Lagrange interpolation* or directly by solving *linear systems* over finite fields.

[4] The symbol $\approxeq$ denotes "reduction by means of polynomial rules"; in order to ease reading, however, we shall use the equality sign everywhere when there is no danger of misunderstanding.

2.1 Algorithms for obtaining polynomial ring systems characterizable by finite matrices.

The algorithm for obtaining the polynomial corresponding to a deterministic or non-deterministic finite-valued logic begins with the construction of truth-tables for each connective that makes up the language of the logical system that will be translated. Thus, we have:

Algorithm 1

Step 1: *Given a fixed logic $\mathcal{L}$, we select a finite field $GF(p^m)$ (for convenient p prime and m a nonzero natural number) such that this field can represent the truth-values of $\mathcal{L}$.*

Step 2: *Select a set D, $D \subseteq GF(p^m)$, for the truth-values taken as distinguished (or designated).*

Step 3: *Define a function that translates formulas of $\mathcal{L}$ on polynomials with coefficients in $GF(p^m)$ with variables in the set X, i.e:*

$$* : For_{\mathcal{L}} \rightarrow GF(p^m)[X]$$

It is proved in [8] that any finite function can be expressed by means of polynomials over suitable finite fields. Namely:

Theorem 2.1 *(Representation of finite functions in $GF(p^n)$): Let A be any finite set with cardinality $|A| = k$, $f : A^m \rightarrow A$ be any function with m arguments on A and $GF(p^n)$ a Galois field with $k \leq p$. Then f can be represented as a polynomial function in $GF(p^n)[x_1, ..., x_m]$.*

In order to determine polynomials by solving linear systems, we recall that the definition of a finite-valued logic depends on a finite number n, $n \geq 2$, of truth-values (e.g., if $n = 2$ we will be dealing with a bivalued system, if $n = 3$ with a three-valued system, and so on). By expressing these truth-values in the form of matrices, all possible logical connectives can be defined.

In this paper, we will limit ourselves to the unary and binary truth-tables representing the great majority of logical connectives in finite-valued systems (but of course the case of k-ary connectives can be treated similarly). For a two-valued logic, all possible truth-tables are expressed by the following polynomials with coefficients in the field $\mathbb{Z}_2$:

$$p(x) = ax + b$$
$$q(x, y) = axy + bx + cy + d$$

In a three-valued logic, similarly, all unary and binary connectives are expressed by the following polynomials, respectively, with coefficients in the field $\mathbb{Z}_3$:

$$p(x) = ax^2 + bx + c$$
$$q(x, y) = ax^2y^2 + a'x^2y + a''x^2y^0 + bxy^2 + b'xy + b''xy^0 + cx^0y^2 + c''x^0y + c''x^0y^0$$

In n-valued systems, all unary and binary connectives will be defined, respectively, by polynomials with coefficients in the field $GF(p^m)$ (for the least p^m, p prime and m a nonzero natural number, such that $n \leq p^m$):

$$p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_0x^0 \text{ and}$$
$$q(x,y) = a_{n-1,n-1}x^{n-1}y^{n-1} + \ldots + a_{i,j}x^iy^j + \ldots + a_{0,0}x^0y^0$$

By assigning values to $q(x,y)$, in the case of a binary connective for instance, we obtain a linear system with n^2 equations and n^2 unknowns, which, as the above-mentioned theorem shows, has a single unique solution.

Equivalently, as shown in [8], polynomials can be determined by Lagrange interpolation, a well-known general method of numerical analysis that enables one to derive a polynomial given a number of points in the polynomial, but here restricted to finite cases.

Both methods (based on the above-mentioned Theorem 2.1 and in Theorem 2.2 below) guarantee the correction of the PoLCa software package described in Section 5.

In the case of systems with non-deterministic matrices the procedure is similar, but with a slight distinction: non-deterministic matrices can be seen as the familiar finite matrices, with the difference that entries may have a set of truth-values, instead of a single truth-value. In the entries where there is a set of truth-values, a new polynomial is introduced to represent such non-deterministic entries. The new polynomials will be defined with coefficients in the Galois field $GF\left(p^m\right)[X \cup X']$, where X' denotes a new set of variables, called hidden variables.

An immediate generalization of Theorem 2.1 applies to non-deterministic matrices.

Theorem 2.2 *(Representation of finite deterministic and non-deterministic functions over Galois Fields): Let $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$ be a finite non-deterministic matrix according to definition 3.1. Then $\mathcal{M}$ can be represented as a polynomial with coefficients in an appropriate Galois field.*

Proof. If each function $\widetilde{\diamond} : \mathcal{V}^n \to 2^{\mathcal{V}} - \{\emptyset\}$ returns only unitary sets (i.e., singletons), this is just a deterministic function and the reasoning is the same as that in Theorem 2.1.

Otherwise, the function $\widetilde{\diamond}$ returns, for an input in $\mathcal{V}$, a set of functions over a non-empty set C in $2^{\mathcal{V}} - \{\emptyset\}$.

But the class of such functions has cardinality $|C|^{|\mathcal{V}|}$, and can be expressed as polynomials over a suitable field $GF(p^n)$. In this case the class of all functions with input $\mathcal{V}^m$ and output V x $GF(P^n)[X']$ are also expressed in terms of polynomials, by applying the result for deterministic functions presented in Theorem 2.1. □

Hidden variables (or more generally hidden terms) are extra algebraic variables, distinct from those associated with propositional variables, supposed to take values in the same field. In this way, the non-deterministic character of the semantics is captured by the assignment of truth-values to the hidden variables. The presence of hidden variables in the polynomial corresponding to a specific formula, indicates

that the truth-values of that formula do not functionally depend on the truth-values of its propositional variables.

It is important to note that the maximum number of hidden variables needed to represent non-deterministic entries in a finite non-deterministic matrix $\widetilde{\diamond} : \mathcal{V}^n \to 2^{\mathcal{V}} - \{\emptyset\}$ is, in the worst case (where all entries are non-deterministic), bounded by $\mathcal{V}^n$. That is, the maximal number of hidden variables to be introduced is the number of truth-values $\mathcal{V}$ at power n, where n is the arity of the matrix. As those parameters are fixed, there is no danger of combinatorial explosion.

As a motivation, we will first define a PRC for the Classical Propositional Calculus (CPL), and then extend it to the non-deterministic systems. In both cases, formulas are translated into polynomials over the field $\mathbb{Z}_2$ ($\mathbb{Z}_2[X]$ in CPC and $\mathbb{Z}_2[X \cup X']$ in non-deterministic matrices) and the only distinguished truth-value is 1. In this case, elements of the form $x + x$ reduce to 0 and elements of the form $x.x$ reduce to x.

Definition 2.3 (PRC for CPL)
Let For_{CPL} be the set of well-formed formulas of CPL, and let $X = \{x_{p_1}, x_{p_2}, ...\}$ be a set of algebraic variables. The PRC for CPL is determined by the translation function $* : For_{CPL} \to \mathbb{Z}_2[X]$ defined by:

- $(p_i)^* = x_i$ for each atomic variable p_i
- $(\neg\alpha)^* = 1 + (\alpha)^*$
- $(\alpha \wedge \beta)^* = (\alpha)^* \cdot (\beta)^*$
- $(\alpha \vee \beta)^* = (\alpha)^* \cdot (\beta)^* + (\alpha)^* + (\beta)^*$
- $(\alpha \to \beta)^* = (\alpha)^* \cdot (\beta)^* + (\alpha)^* + 1$

An illustrative example of a derivation in CPL is the following:

Example 2.4 $\not\approx_{CPL} (p \vee \neg p)$.
$(p \vee \neg p)^* = p^* + (\neg p)^* + p^* + (\neg p)^* =$
$= x(x + 1) + x + x + 1 =$
$= x.x + x + x + x + 1$
$= x + x + x + x + 1$
$= 0 + 0 + 1$
$= 1$

3 Non-deterministic semantics via polynomials

In [2], and in other papers, A. Avron proposes the concept of non-deterministic truth-values through a generalization of the usual matrices, where the truth-value of a given complex formula is non-deterministically selected within a set of options. This non-deterministic approach, when applied to many-valued matrices, extends the notion of truth-functionality, at least when the non-deterministic character of the choices in question is finitely bounded.

The question of analyticity in logic refers to the property that, in order to deter-

mine whether ϕ follows from Δ, it always suffices to check only information concerning subformulas of $\Delta \cup \{\phi\}$. Several semantic approaches, as the possible-translations semantics (cf. [4], also in [9]) and the non-deterministic matrices, permit to separate the notion of truth-functionality from the notion of analyticity: some semantics can be analytic without being necessarily truth-functional. Non-deterministic matrices constitute a slight generalization of many-valued matrices, but are applicable to a wider range of logics that are not characterizable by means of ordinary finite matrices. Although the notions of analyticity in philosophy and logic do not coincide, they are certainly connected: in his book *The Roots of Reference*, of 1974, W. V. O. Quine defended a broader account for the notion of analyticity in terms of what we may call "the learning process". Quine even suggested ternary non-functional matrices (perfectly non-deterministic, from our perspective, cf. [16], pp. 76-77), curiously anticipating non-deterministic matrices [5].

Definition 3.1 A non-deterministic matrix (Nmatrix for short) for a propositional language $\mathcal{L}$ is a tuple $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where:

- $\mathcal{V}$ is a non-empty set of *truth-values*;
- $\mathcal{D}$ is a non-empty proper subset of $\mathcal{V}$, i.e, $\mathcal{D} \subseteq \mathcal{V}$.
- For every n-ary connective $\diamond$ of $\mathcal{L}$, $\mathcal{O}$ includes a corresponding n-ary function $\widetilde{\diamond}$ from $\mathcal{V}^n$ into $2^{\mathcal{V}} - \{\emptyset\}$.

  We say that $\mathcal{M}$ is (in)finite if $\mathcal{V}$ is also.

Definition 3.2 Let W be the set of formulas of $\mathcal{L}$. A (legal) valuation in an Nmatrix $\mathcal{M}$ is a function $v : W \to \mathcal{V}$ that satisfies the following condition for every n-ary connective $\diamond$ of $\mathcal{L}$ and $\psi_1, ..., \psi_n \in W$:

$$v(\diamond(\psi_1, ..., \psi_n)) \in \widetilde{\diamond}(v(\psi_1), ..., v(\psi_n)).$$

As expected, an ordinary (deterministic) matrix is identified with an Nmatrix whose functions in $\mathcal{O}$ always return singletons. In this way, non-deterministic matrices constitute a genuine generalization of the deterministic ones. Two examples bellow illustrate how such semantics generalize the classical case.

Example 3.3 Consider the language $\mathcal{L} = \langle \wedge, \vee, \rightarrow, \neg \rangle$, whose operators $(\wedge, \vee, \rightarrow)$ are interpreted classically, while negation allows the law of contradiction but does not necessarily support the law of excluded middle. These conditions define the following Nmatrices $\mathcal{M}^2 = (\mathcal{V}, \mathcal{D}, \mathcal{O})$ for $\mathcal{L}$, where $\mathcal{V} = \{t, f\}$, $\mathcal{D} = \{t\}$ and $\mathcal{O}$ are

[5] We thank Prof. Marcello D'Agostino (University of Ferrara) for this observation; see [13]

given by:

	$\widetilde{\neg}$
t	$\{f\}$
f	$\{t, f\}$

		$\widetilde{\vee}$	$\widetilde{\wedge}$	$\widetilde{\rightarrow}$
t	t	$\{t\}$	$\{t\}$	$\{t\}$
t	f	$\{t\}$	$\{f\}$	$\{f\}$
f	t	$\{t\}$	$\{f\}$	$\{t\}$
f	f	$\{f\}$	$\{f\}$	$\{t\}$

Classical negation can be here defined in $\mathcal{M}^2$ by: $\sim \psi = \psi \rightarrow \neg\psi$. In terms of truth-tables we have:

	$\widetilde{\neg}$	$\widetilde{\sim}$
t	$\{f\}$	$\{f\}$
f	$\{t, f\}$	$\{t\}$

Thus the full propositional classical logic PRC can be defined within this logic.

Example 3.4 Consider the language $\mathcal{L} = \langle \wedge, \vee, \rightarrow, \neg \rangle$, whose operators $(\wedge, \vee, \rightarrow)$ are interpreted classically, while negation allows the law of contradiction but does not necessarily support the law of excluded middle. These conditions define the following Nmatrices $\mathcal{M}^2 = (\mathcal{V}, \mathcal{D}, \mathcal{O})$ for $\mathcal{L}$, where $\mathcal{V} = \{t, f\}$, $\mathcal{D} = \{t\}$ and $\mathcal{O}$ is given by:

	$\widetilde{\neg}$
t	$\{f\}$
f	$\{t, f\}$

		$\widetilde{\vee}$	$\widetilde{\wedge}$	$\widetilde{\rightarrow}$
t	t	$\{t\}$	$\{t\}$	$\{t\}$
t	f	$\{t\}$	$\{f\}$	$\{f\}$
f	t	$\{t\}$	$\{f\}$	$\{t\}$
f	f	$\{f\}$	$\{f\}$	$\{t\}$

Classical negation can, again, be defined in $\mathcal{M}^2$ by: $\sim \psi = \psi \rightarrow \neg\psi$, and thus PRC can be now be defined within this logic. In terms of truth-tables we have:

	$\widetilde{\neg}$	$\widetilde{\sim}$
t	$\{f\}$	$\{f\}$
f	$\{t, f\}$	$\{t\}$

4 A three-valued polynomial version of the Logic of Formal Inconsistency mbC

As shown in [2], an Nmatrix for the mbC system is given by the triple $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where:

- $\mathcal{V} = \{f, t, I\} = \{0, 1, 2\}$;

- $\mathcal{D} = \{t, I\} = \{1, 2\}$;
- $\mathcal{O}$ are operations defined by:

$\widetilde{\vee}$	0	2	1
0	$\{0\}$	$\{1,2\}$	$\{1,2\}$
2	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
1	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$

$\widetilde{\wedge}$	0	2	1
0	$\{0\}$	$\{0\}$	$\{0\}$
2	$\{0\}$	$\{1,2\}$	$\{1,2\}$
1	$\{0\}$	$\{1,2\}$	$\{1,2\}$

$\widetilde{\rightarrow}$	0	2	1
0	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
2	$\{0\}$	$\{1,2\}$	$\{1,2\}$
1	$\{0\}$	$\{1,2\}$	$\{1,2\}$

$\widetilde{\neg}$	0	2	1
	$\{1,2\}$	$\{1,2\}$	$\{0\}$

$\widetilde{\circ}$	0	2	1
	$\{0,1,2\}$	$\{0\}$	$\{0,1,2\}$

The procedure for obtaining the corresponding polynomials, in this case in the field $\mathbb{Z}_3$, can be simplified by defining two-valued auxiliary tables with entries 1 (to denote "presence") and 0 (to denote "absence") of the sets of truth-values. Thus, in the places where there are more than one truth-value the entries are 1, and 0 otherwise. From these auxiliary tables we obtain auxiliary polynomials which will be submitted to necessary changes in order to represent the non-deterministic tables.

The auxiliary tables in this case are:

$\widetilde{\vee}$	0	2	1
0	$\{0\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$

$\widetilde{\wedge}$	0	2	1
0	$\{0\}$	$\{0\}$	$\{0\}$
2	$\{0\}$	$\{1\}$	$\{1\}$
1	$\{0\}$	$\{1\}$	$\{1\}$

$\widetilde{\rightarrow}$	0	2	1
0	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{0\}$	$\{1\}$	$\{1\}$
1	$\{0\}$	$\{1\}$	$\{1\}$

$\widetilde{\neg}$	0	2	1
	$\{1\}$	$\{1\}$	$\{0\}$

$\widetilde{\circ}$	0	2	1
	$\{1\}$	$\{0\}$	$\{1\}$

4.1 Polynomial representation for the disjunction operator in mbC

As an example, we compute here in detail the disjunction operator for mbC as expressed in non-deterministic matrices. As noted above, an Nmatrix for the mbC system is three-valued, and thus the general polynomials that will represent the unary and binary connectives are given by:
(i) $p(x) = ax^2 + bx + c$.
(ii)$p(x, y) = ax^2y^2 + a'x^2y + a''x^2y^0 + bxy^2 + b'xy + b''xy^0 + cx^0y^2 + c'x^0y + c''x^0y^0$.

As mentioned above, the disjunction operator for mbC in terms of a Nmatrix is given by:

$\widetilde{\vee}$	0	2	1
0	{0}	{1,2}	{1,2}
2	{1,2}	{1,2}	{1,2}
1	{1,2}	{1,2}	{1,2}

$\widetilde{\vee}$	0	2	1
0	{0}	{1}	{1}
2	{1}	{1}	{1}
1	{1}	{1}	{1}

By analyzing the auxiliary table we see that
$p(0, 0) = 0$;
$p(0, 2) = p(0, 1) = p(2, 0) = p(2, 2) = p(2, 1) = p(1, 0) = p(1, 2) = p(1, 1) = 1$.

Now, by plugging such values in the general polynomial $p(x, y) = ax^2y^2 + a'x^2y + a''x^2y^0 + bxy^2 + b'xy + b''xy^0 + cx^0y^2 + c'x^0y + c''x^0y^0$, we obtain a linear system of nine equations in the unknowns $a, a', a'', b, b', b'', c, c', c''$.

The system quickly gives $c'' = 0$, $c' = 0$, $c = 1$, $b'' = 0$ and $a'' = 1$, and reduces to the linear system of the following four equations:

$$(1) \qquad \begin{cases} a + 2a' + 2b + b' + 2 = 1 \\ a + a' + 2b + 2b' + 2 = 1 \\ a + 2a' + b + 2b' + 2 = 1 \\ a + a' + b + b' + 2 = 1 \end{cases}$$

Solving the system, the auxiliary polynomial for disjunction is:

$$p(x, y) = 2x^2y^2 + x^2 + y^2$$

It is now essential to recall that the auxiliary matrix was obtained by replacing the values $\{1, 2\}$ by $\{1\}$ in the Nmatrix of the disjunction operator. Now that the auxiliary polynomial has been determined, we need to rescue the characteristic of the Nmatrix, and this will be done by using some extra (hidden) variables.

5 Hidden variables in a general three-valued scenario

The idea here is that non-deterministic sets of truth-values (that is, non-empty subsets of $\{0, 1, 2\}$) can be represented by specific polynomials in new, extra variables,

which we call *hidden variables*.

As an example, note that the polynomial x^2 is constrained into the set $\{0,1\}$, since for any $x \in \{0,1,2\}$ we have that $x^2 \in \{0,1\}$, or in other words:

- If $x = 0$ then $0^2 = 0$.
- If $x = 1$ then $1^2 = 1$.
- If $x = 2$ then $2^2 = 4 \equiv 1 \pmod 3$.

There are obviously other restrained polynomials for the same set of truth-values (actually, as many as there are surjective functions from $\{0,1,2\}$ into $\{0,1\}$), but for our purposes it is sufficient to use any one of them.

In the same way, the polynomial x^2+1 is constrained into the set $\{1,2\}$ (and the same occurs for the polynomial 2. $\left(x^2+1\right)$.) and the polynomial $2x^2$ is constrained into the set $\{0,2\}$.

In this way, the set of truth-values $\{1,2\}$ in the Nmatrix can be expressed by the polynomial $\left(x'^2+1\right)$, using a new (hidden) variable x'.

Thus we have to multiply the auxiliary polynomial obtained above by the new $\left(x'^2+1\right)$, and the result is the same as expressing in polynomial terms the table below, whose entries are polynomial themselves:

$\widetilde{\vee}$	0	2	1
0	$\{0\}$	x'^2+1	x'^2+1
2	x'^2+1	x'^2+1	x'^2+1
1	x'^2+1	x'^2+1	x'^2+1

The reader may notice that our choice of representing in the auxiliary table the set $\{1,2\}$ by 1 is arbitrary (but convenient). If we had represented $\{1,2\}$ by 2, we would just replace the polynomial $\left(x'^2+1\right)$ by $2(x'^2+1)$ (taking into account the field arithmetic).

Therefore the resulting polynomial for the disjunction operator in mbC is finally given by:

$$p_\vee(x,y) = \left(2x^2y^2+x^2+y^2\right)\left(x'^2+1\right)$$

Now, reasoning in the same line for the remaining operators, we obtain:
$p_\vee(x,y) = \left(2x^2y^2+x^2+y^2\right)\left(x'^2+1\right)$;
$p_\wedge(x,y) = (x.y)\left(x'^2+1\right)$;
$p_\rightarrow(x,y) = \left(x^2y^2+2x^2+1\right)\left(x'^2+1\right)$;
$p_\neg(x,y) = \left(x^2+x+1\right)\left(x'^2+1\right)$;
$p_\circ(x,y) = \left(x^2+2x+1\right)(x')$.

It is important to emphasize that polynomial versions for the LFIs (see [9] for such logics), expressed by polynomials over the field $\mathbb{Z}_2$, can also be defined for binary non-truth-functional semantics, as done in [7]. The resulting polynomials (over $\mathbb{Z}_2$) are different from those shown here (over $\mathbb{Z}_3$). In a certain sense, the Nmatrices recover part of the truth-functionality, which is lost in the binary semantics, and the polynomial representations neatly express such nuances. For more details, see [7].

Note that the translations $*$ translate formulas into polynomials of degree n within the ring $\mathbb{Z}_2[X \cup X']$, where $X = \{x_p : p \in For_{\mathcal{L}}\}$ and $X' = \{x'_p : p \in For_{\mathcal{L}}\}$. In this way, polynomials can be regarded as syntactical elements, whose semantic part are the interpretations in $I : \mathbb{Z}_2[X \cup X'] \to \mathbb{Z}_2$.

Theorem 5.1 *For each propositional valuation v, there is an operation (as defined in $*$) and ring homomorphism $I : \mathbb{Z}_2[X] \to \mathbb{Z}_2$, such that:*

$$v(\alpha) = I(\alpha^*), \textit{ i.e.:}$$
$$v(\alpha) = 1 \textit{ iff } I(\alpha^*) = 1 \in \mathbb{Z}_2.$$

And, consequently:

$$v(\alpha) = 0 \textit{ iff } I(\alpha^*) = 0 \in \mathbb{Z}_2.$$

Proof. Let $v : For_{\mathcal{L}} \to \{0, 1\}$ be a valuation. Define,
$I(x_p) = 1$ iff $v(p) = 1$;
$I(x_p) = 0$ iff $v(p) = 0$.

In less formal terms, we have:
$I(x_p) = v(p)$.

It only remains to demonstrate that $v(\alpha) = I(\alpha^*), \alpha \in For$. Therefore, by induction on the complexity of the formulas, we have:

- For **negation**:
  $I(\neg\alpha)^* = 1 \Leftrightarrow I(\alpha)^* + 1 = 1 \Leftrightarrow_{\mathbb{Z}_2} I(\alpha)^* = 0 \Leftrightarrow_{HI} v(\alpha) = 0 \Leftrightarrow v(\neg\alpha) = 1.$

  $I(\neg\alpha)^* = 0 \Leftrightarrow I(\alpha)^* + 1 = 0 \Leftrightarrow_{\mathbb{Z}_2} I(\alpha)^* = 1 \Leftrightarrow_{HI} v(\alpha) = 1 \Leftrightarrow v(\neg\alpha) = 0.$

- For **conjunction**:
  $I(\alpha \wedge \beta)^* = I(\alpha)^*.I(\beta)^* = 1 \Leftrightarrow_{\mathbb{Z}_2} I(\alpha)^* = 1$ and $I(\beta)^* = 1 \Leftrightarrow_{HI} v(\alpha) = 1$ and $v(\beta) = 1 \Leftrightarrow v(\alpha \wedge \beta) = 1$.

  $I(\alpha \wedge \beta)^* = I(\alpha)^*.I(\beta)^* = 0 \Leftrightarrow_{\mathbb{Z}_2} I(\alpha)^* = 0$ or $I(\beta)^* = 0 \Leftrightarrow_{HI} v(\alpha) = 0$ or $v(\beta) = 0 \Leftrightarrow v(\alpha \wedge \beta) = 0$.

- For $(\vee, \rightarrow)$ the proof is analogous.

□

Thus, the formula $\alpha \in \mathcal{L}$ is *satisfiable* if its polynomial translation $\alpha^* \in \mathbb{Z}_2[X]$ is closed within a certain set $\mathbf{D} \in F$ of distinguished truth-values.

Theorem 5.2 *For each valuation v, define an interpretation $I : \mathbb{Z}_2[X \cup X'] \to \mathbb{Z}_2$ as a ring homomorphism such that $v = I_0()^*$, i.e., $v(\alpha) = I(\alpha^*)$. So, for all sentences α in mbC,*

$$v(\alpha) = \begin{cases} \mathbf{1}, \; iff \;\; I(\alpha^*) = 1. \\ \mathbf{0}, \; iff \;\; I(\alpha^*) = 0. \end{cases} \tag{2}$$

Proof. To be found in [15]. □

Based on theorems 5.1 and 5.2, we then conclude that the above polynomials provide a polynomial representation for the correct and complete structure given in [2], section 3. From this, it is clear that our approach yields a correct and complete semantics for mbC (in this case, three-valued and non-deterministic) in terms of polynomials.

There exists however, also a deterministic two-valued semantics for the same calculus mbC. Generally, when there is a many-valued non-deterministic semantics (which we might call "truth-relational") for some logic there will be also a two-valued semi-truth functional. In [9] the authors present a semi-truth-functional semantic for the system mbC; that semantics can also be represented in terms of polynomials, which makes a common base for comparisons.

Let Σ° be the signature formed by $\Sigma^\circ = \{\wedge, \vee, \rightarrow, \neg, \circ\}$ such that $\mathcal{P} = \{p_n : n \in \omega\}$ is the set of atomic formulas and $\circ$ a unary operator. We define, as usual, For° as the set of formulas freely generated by $\mathcal{P}$ over For°.

Let $\mathbf{2} =_{def} \{0, 1\}$ be the set of two truth-values, where 1 denotes the "true" value and 0 denotes the "false" value. An mbC-valuation is any function $v : For^\circ \to 2$ subject to the following clauses:
(v1) $v(\alpha \wedge \beta) = 1$ sse $v(\alpha) = 1$ and $v(\beta) = 1$.
(v2) $v(\alpha \vee \beta) = 1$ sse $v(\alpha) = 1$ or $v(\beta) = 1$.
(v3) $v(\alpha \rightarrow \beta) = 1$ sse $v(\alpha) = 0$ or $v(\beta) = 1$.
(v4) $v(\neg\alpha) = 0$ implies $v(\alpha) = 1$.
(v5) $v(\circ\alpha) = 1$ implies $v(\alpha) = 0$ ou $v(\neg\alpha) = 0$.

A polynomial ring calculus for the mbC system in this semi-truth-functional semantics is defined by the steps below. Let $X = \{x_{p_1}, x_{p_2}, ...\}$ and $X' = \{x_{\alpha_1}, x_{\alpha_2}, ...\}$ disjoint sets of algebraic variables, indexed by propositional variables p_1 and by mbC formulas denoted by α_i, respectively. The variables in X' are the hidden variables in this case. The polynomial ring calculus for mbC with respect to this two-valued semantics is defined by the translation function $*$:

$$* : For_{mbC} \to \mathbb{Z}_2 [X \cup X']$$

such that:

(i) $(p_i)^* = x_i$, for $x_i \in X$, p atomic.

(ii) $(\alpha \wedge \beta)^* = \alpha^* \beta^*$.

(iii) $(\alpha \vee \beta)^* = \alpha^*\beta^* + \alpha^* + \beta^*$.

(iv) $(\alpha \rightarrow \beta)^* = \alpha^*\beta^* + \alpha^* + 1$.

(v) $(\neg\alpha)^* = \alpha^* x_\alpha + 1$, onde x_α is a hidden variable in X'.

(vi) $(\circ\alpha)^* = (\alpha^* (x_\alpha + 1) + 1)\, x_\alpha$, onde x_α is a hidden variable X'.

We have thus the same system (mbC) characterized by two distinct semantics, whose polynomial characterizations have different natures. The use of finite structures of the kind of non-deterministic semantics has the benefit of preserving the advantages of logics with ordinary finite-valued semantics (in particular: decidability and compactness), while it is applicable to a much larger family of logics. Another important point about Nmatrices is that the generalization of the concept of many-valued matrix allows us to provide a finite structure for a logic that is not characterizable by finite (truth-functional) matrices, as in the case of mbC. This automatically provides a decision procedure for mbC, for instance, which is only given by means of the more general notion of possible-translations semantics (see [9]). Although the concept of Nmatrices is but a particular case of the concept of possible-translations semantics, Nmatrices constitute a handier tool for computing the semantics, and its polynomial expression also has this characteristic.

Of course the above method applies for Nmatrices in general; in particular all the LFIs that may be characterizable by non-deterministic matrices, although having non- finite-valued semantics, can be treated similarly.

As a final comment on the method, there are some natural connections between our polynomial semantics and the relational semantics, as introduced by J. M. Dunn in [14] and studied in several other papers, notably in [3]. It should be clear that our restrictions to finite structures in this paper are not essential. Abstract characterizations of logics by means of polynomials over arbitrary fields, or vice-versa, characterizations of general structures defined over rings of formal power series interpreted as logics, are a next step in this study.

6 The PoLCa software

The PoLCa software [6] (open source and publicly available) translates sentences of several logics (such as many-valued, paraconsistent, etc.), whose semantics are deterministic (truth-functional) or non-deterministic (controlled non-truth-functional) into polynomials over finite fields, automatizing what has been shown in the above examples. Proofs in such systems, then, reduce to handling polynomials in a natural and intuitive way.

Briefly, given a collection of truth-tables, described by deterministic matrices or

[6] Programmed by Glauber De Bona, designed by Mariana Matulovic and Walter Carnielli.

non-deterministic matrices (Nmatrices) which define logical operators of arbitrary arities, the PoLCa package (Polynomial Ring Calculus Software) computes the polynomials whose integer variables represent the arguments of the logical operators, such that polynomials will simulate all the input-output values of the corresponding (deterministic or non-deterministic) truth-table. The correctness of the program is guaranteed by Theorems 2.1 and 2.2.

The program input is given by a text file, which specifies the matrices of the operators under interest. The first line (header) file must specify, in order, the number of truth-values, the number of logical operators whose matrices are specified and the arity of each of the operators. Such values should be specified with natural numbers separated by spaces. For example, if we are interested in two operators, respectively, ternary and unary, with five truth-values, the first line of the input file should contain:

5 2 3 1 (Five truth-values, two operators, arity of operators, respectively, three and one.)

In a logic with n truth-values, the values will be represented by natural numbers 0,1,2 ..., n-1 but taken as elements of the smallest Galois Field, $GF(p^m)$ such that $n \leq p^m$. For example, consider a truth-table of a unary operator o(x):

x	0	1	2
$o(x)$	1	0	2

It would be represented, in the program, by the line:

1 0 2

To represent sets of values (non-deterministic) tables, just put the elements of the set in braces separated by commas. For example, if for certain values of its arguments an operator can take the values 1 or 2, they should be placed in the position they appear in the N-matrix.

The output of the program is purely textual. For each operator whose (deterministic or non-deterministic) truth-table is specified in the input, a polynomial is returned separately. Variables are letters in the alphabetical order reflecting the order of the arguments implicit in the specification of the truth-tables of the input, as explained above. The output string simply lists the coefficients, including zero, in the reverse order in which the polynomial is presented.

For more details, see:
http://marianamatulovic.wix.com/polyringcalculus

Acknowledgement

This research was financed by FAPESP (Thematic Project LogCons 2010/51038-0, Brazil) and by individual research grants from The National Council for Scientific and Technological Development (CNPq), Brazil. Additional sponsorship has been provided by CPAI-UnB, Brazil.

References

[1] Agudelo, J. C., Carnielli, W. A., *Polynomial Ring Calculus for Modal Logics: a new semantics and proof method for modalities*, The Review of Symbolic Logic. **4** (2011), 150–170, URL: http://doi:10.1017/S1755020310000213.

[2] Avron, A., *Non-deterministic Semantics for Logics with a Consistency Operator*, Journal of Approximate Reasoning. **45** (2007), 271–287.

[3] Bimbó, K., and Dunn, J. M.,"Generalized Galois Logics: Relational Semantics of Nonclassical Logical Calculi", CSLI Lecture Notes- CSLI Publications, 2008.

[4] Carnielli, W. A., *Possible-translations semantics for paraconsistent logics*. In: Frontiers in Paraconsistent Logic: Proceedings of the I World Congress on Paraconsistency, Ghent, 1998, pp. 15972 , edited by D. Batens et al., Kings College Publications, 2000.

[5] Carnielli, W. A., *A polynomial proof system for Łukasiewicz logics*, Second Principia International Symposium. (2001), 6–10.

[6] Carnielli, W. A., *Polynomial ring calculus for many-valued logics*, Proceedings of the 35th International Symposium on Multiple-Valued Logic, IEEE Computer Society. Calgary, Canada (2005), 20–25.

[7] Carnielli, W. A., *Polynomial Ring Calculus for Logical Inference*, CLE e-Prints. **5** (2005), 1–17. URL: ftp://ftp.cle.unicamp.br/pub/e-prints/vol.5,n.3,2005.pdf.

[8] Carnielli, W. A., *Polynomizing: Logic Inference in Polynomial Format and the Legacy of Boole*, Model-Based Reasoning in Science, Technology, and Medicine, Magnani, L., and Li, P., Eds., Springer publisher. **64** (2007), 349–364.

[9] Carnielli, W. A., Coniglio, M. E., and Marcos, J., *Logics of Formal Inconsistency*, Handbook of Philosophical Logic, Gabbay, D., and Guenthner, F., Eds **14** (2007), 15–107

[10] Carnielli, W. A., *Formal polynomials and the laws of form*, The Multiple Dimensions of Logic, Béziau, J. Y., Costa-Leite, A., Eds. **54** (2009), 2002–2012.

[11] Carnielli, W. A., *Formal polynomials, heuristics and proofs in logic*, Logical Investigations, Karpenko, A. S., Ed., Institute of Philosophy- Russian Academy of Sciences Publisher **16** (2010), 280–294.

[12] Carnielli, W. A., *Proofs by handling polynomials: a tool for teaching logic and metalogic*, Proceedings of the Third International Congress on Tools for Teaching Logic, Salamanca, Spain. (2011), 1–3.

[13] D'Agostino, M., *Analytic inference and the informational meaning of the logical operators*, Logique et Analyse, in print (2013).

[14] Dunn, J. M., *Intuitive semantics for first-degree entailments and coupled trees*, Philosophical Studies **29** (1976), 149–169

[15] Matulovic, M., "Proofs in the *Algibeira*: Polynomials as a Universal Method of Proof", Ph.D. Thesis State University of Campinas (UNICAMP), Brazil, 2013.

[16] Quine, W. V. O., "The Roots of Reference", Open Court, 1973.