Available on line at www.rac.es/racsam
Computational Sciences
Survey

**REVISTA DE LA REAL ACADEMIA DE CIENCIAS**
**EXACTAS, FISICAS Y NATURALES.**
SERIE A: MATEMATICAS
Madrid (España / Spain)

RACSAM **104** (1), 2010, 19–40. DOI:10.5052/RACSAM.2010.04

# An algebraic approach to rule based expert systems

**Eugenio Roanes-Lozano, Luis M. Laita, Antonio Hernando and Eugenio Roanes-Macías**

**Abstract** This article presents a survey of the authors' research on knowledge extraction and verification of Rule Based Expert Systems (RBES) using algebraic inference engines and based on Gröbner bases theory. A shell, including a graphic user interface and inference engines for different logics (both classic and modal multi-valued) as well as in different computer algebra systems, is also presented here. The shell distinguishes three levels: at the lower level, we provide the computer algebra system code of the algebraic inference engines; at the intermediate level, the RBES developer has to detail the rules and integrity constraints of a certain RBES; and, finally, at the upper level, the final user deals with a simple GUI, where he can perform knowledge extraction or verify the RBES, after choosing the logic and inputing a consistent set of facts. We believe that this shell can be really useful for teaching and quick RBES design.

## Una aproximación algebraica a los sistemas expertos basados en reglas

**Resumen.** Este artículo presenta una panorámica de la línea de investigación de los autores en extracción de conocimiento y verificación de Sistemas Expertos Basados en Reglas (RBES) usando motores de inferencia algebraicos y basada en la teoría de bases de Gröbner. Se presenta también una *shell*, que incluye una interfaz gráfica de usuario y motores de inferencia para distintas lógicas (tanto clásicas como modales multivaluadas) y en distintos sistemas de cómputo algebraico. La *shell* distingue tres niveles: en el más bajo proporcionamos el código del motor de inferencia para el sistema de cómputo algebraico elegido; en el intermedio el desarrollador del RBES tiene que detallar las reglas y las restricciones de integridad de un cierto RBES; y, finalmente, en el nivel superior, el usuario final trata con una sencilla interfaz gráfica de usuario, en la que puede llevar a cabo extracción de conocimiento o verificar el RBES, después de elegir la lógica y de introducir un conjunto consistente de hechos. Creemos que esta *shell* puede ser realmente útil para la enseñanza y para el rápido diseño de RBES.

Submitted by David Ríos Insua.
*Received*: August 2, 2009. *Accepted*: October 19, 2009
*Keywords*: Rule based expert systems, logic and symbolic computing, Groebner bases
*Mathematics Subject Classifications*: 03B05, 13P10, 03B80

## 1 Introduction

This article presents a survey of the authors' research on knowledge extraction and verification of Rule Based Expert Systems (RBES) using algebraic inference engines. These inference engines are based on the use of Gröbner bases. The leader of the research team is Luis Laita, and the first works on the topic are dated in the early 90's, although he had already been working in the field using other techniques.

The present paper is self-contained, in such way that introductory well-known sections 2, 5, 7may be skipped by an acquainted reader, while the other sections convey a view of new results and approaches by the authors. The article begins with a brief introduction to lattices, lattice orders, Boolean algebras and Boolean rings (Section 2). It is followed by the construction of the polynomial model for Boolean logic due to these authors (Sections 3 and 4). Then a brief introduction to modal multi-valued logics can be found (Section 5). Section 6 introduces the main result of this theory, relating to be a tautological consequence with a polynomial ideal membership. A brief introduction to RBES is included afterwards (Section 7). Section 8 details the adaptation of the algebraic model for logic to RBES. In Section 9 it is shown how two levels of inconsistency can be distinguished in the multi-valued case (including an interpretation in algebraic geometry). Section 10 introduces how (in RBES whose underlying logic is classic Boolean logic), given a set of facts, it is possible to find out which new fact can make a certain goal be inferred, just computing a Gröbner basis. Implementations in the computer algebra systems (CAS) *CoCoA* and *Maple* are included in Section 11. Finally, a GUI for keeping the CAS hidden from the final user of the RBES is presented in Section 12.

In view that quite a number of topics are treated, many proofs are omitted for the sake of brevity (in fact, some of the proofs are very long, for instance, the one connecting the concept of being tautological consequence in multi-valued logic with an ideal membership in a polynomial residue class ring). Anyway, references where the reader can find details on the topics treated can be found along the paper.

## 2 An introduction to Boolean algebras and Boolean rings

An elementary introduction to Boolean algebras may be found in [21]. The proofs omitted in this section can be found in [18, 28] (these references also extend both Sections 3 and 4). For a deeper study of Boolean algebras see, for instance, [10, 11, 12, 22, 36].

### 2.1 Lattices and lattice orders

**Definition 1** *A set $L$ where two binary operations $\sqcup$ and $\sqcap$ are defined is said to be a* lattice *if and only if both operations are commutative and associative and the absorption laws holds, i.e., if and only if for all elements $a$, $b$, $c \in L$:*

$$\begin{aligned} a \sqcup b &= b \sqcup a & \qquad a \sqcap b &= b \sqcap a \\ a \sqcup (b \sqcup c) &= (a \sqcup b) \sqcup c & \qquad a \sqcap (b \sqcap c) &= (a \sqcap b) \sqcap c \\ a \sqcup (b \sqcap a) &= a & \qquad a \sqcap (b \sqcup a) &= a \end{aligned}$$

**Proposition 1** *If $(L, \sqcup, \sqcap)$ is a lattice, both operations verify idempotency, i.e., for every element $a \in L$:*

$$a \sqcup a = a \qquad a \sqcap a = a$$

Surprisingly, idempotency, although redundant, is often required as a fourth condition for a structure to be a lattice.

**Definition 2** *A non-strict partial order defined over the set $L$ is said to be a* lattice order *if and only if every pair of elements of $L$ has a unique infimum and a unique supremum.*

**Proposition 2** *From the two lattice operations, $\sqcup$ and $\sqcap$, a lattice order $\sqsubseteq$ can be defined: for all elements $a$, $b \in L$:*

$$a \sqsubseteq b \Leftrightarrow a \sqcup b = b \; (\Leftrightarrow a \sqcap b = a)$$

*Conversely, from a lattice order $\sqsubseteq$, the two operations of a lattice can be defined: for all elements $a$, $b \in L$:*

$$a \sqcup b = \sup_{\sqsubseteq}(a, b)$$
$$a \sqcap b = \inf_{\sqsubseteq}(a, b)$$

**Example 1** *Let us consider the lattice of "subsets of $E$": $(\mathcal{P}(E), \cup, \cap)$. From $(\mathcal{P}(E), \cup, \cap)$, $(\mathcal{P}(E), \subseteq)$ is obtained. Conversely, from $(\mathcal{P}(E), \subseteq)$, $(\mathcal{P}(E), \cup, \cap)$ is obtained.*

## 2.2 Boolean algebras

**Definition 3** *A lattice $(L, \sqcup, \sqcap)$ is said to be* distributive *if and only if both operations are distributive w.r.t. the other operation, i.e., if and only if for all elements $a$, $b$, $c \in L$:*

$$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c) \qquad a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$$

**Definition 4** *A lattice is said to be* bounded *if and only if it has a greatest element (top), $\underline{g}$, and least element (bottom), $\underline{l}$.*

**Definition 5** *A bounded lattice $(L, \sqcup, \sqcap)$ is said to be* complemented *if and only if for every element $a \in L$, there is an element $a' \in L$ such that:*

$$a \sqcup a' = \underline{g} \qquad a \sqcap a' = \underline{l}$$

The top and bottom of a lattice are traditionally denoted 0 and 1. However this seems not to be an adequate notation, as the top and bottom of $(L, \sqcup, \sqcap)$ and those of $(L, \sqcap, \sqcup)$ are exchanged.

**Definition 6** *A* Boolean algebra *is a distributive and complemented lattice.*

**Example 2** $(\mathcal{P}(E), \cup, \cap, \, ')$ *is a Boolean algebra:*

- *$\cup$ is distributive w.r.t. $\cap$ and viceversa.*
- *The least and the greatest of the lattice (for $\subseteq$) are $\emptyset$ and $E$, respectively. Moreover, the usual complement of a set, $'$, works as expected:*

$$\forall A \in \mathcal{P}(E) : \; A \cup A' = E \,; \; A \cap A' = \emptyset$$

Let us underline that being *distributive* is completely independent from being *complemented*, as shown below.

**Example 3** *The lattice of the linear subspaces (also denoted "vector subspaces") of $\mathbb{R}^2$, the sum of linear subspaces and intersection is a complemented, but not distributive, lattice:*

- *the complement of a line is any other (different) line, and the complement of the whole plane is $\{\overline{0}\}$,*
- *if $R$, $S$, $T$ are different lines, $R + (S \cap T) \neq (R + S) \cap (R + T)$).*

**Example 4** *The lattice* $(\mathbb{N}, \mathrm{lcd}, \mathrm{gcd})$ *(where* lcd *and* gcd *stand for "least common multiple" and "greatest common divisor", respectively) is distributive, but it is not complemented:* 1 *and* 0 *are the least and greatest of the lattice (respectively), but, for instance,* 3 *has no complement.*

**Example 5** *The lattice of the convex subsets of the plane, the convex union $\overset{*}{\cup}$ (i.e., the least convex set that contains the usual union) and intersection is a lattice, but it is neither distributive nor complemented:*

- *for instance, if $A$, $B$, $C$ are three disjoint and aligned squares of equal size and $A$ is the one in the middle $A \cap (B \overset{*}{\cup} C) = A$, meanwhile $(A \cap B) \overset{*}{\cup} (A \cap C) = \emptyset \overset{*}{\cup} \emptyset = \emptyset$,*
- *there is no convex subset of the plane that behaves as the complement of a line.*

## 2.3 Boolean rings

**Definition 7** *A* Boolean ring *is a ring with unit such that the second operation is idempotent.*

Let us include below some well-known results that will be useful when obtaining the polynomial model for Boolean logic. Let us denote by $(L, \Delta, \sqcap)$ a general Boolean ring, by $\underline{0}$ its neutral element, by $'$ the symbol for the opposite and by $\underline{1}$ its unit.

**Proposition 3** *In any Boolean ring $(L, \Delta, \sqcap)$ any element is its own opposite.*

PROOF. For any elements $a, b \in L$:

$$\begin{aligned}(a \,\Delta\, b) \,\Delta\, \underline{0} = a \,\Delta\, b = (a \,\Delta\, b) \sqcap (a \,\Delta\, b) &= (a \sqcap a) \,\Delta (a \sqcap b) \,\Delta (b \sqcap a) \,\Delta (b \sqcap b) \\ = a \,\Delta ((a \sqcap b) \,\Delta (b \sqcap a)) \,\Delta\, b &= (a \,\Delta\, b) \,\Delta ((a \sqcap b) \,\Delta (b \sqcap a))\end{aligned}$$

but $(L, \Delta)$ is a group, so the opposite is unique, and consequently: $\underline{0} = (a \sqcap b) \,\Delta (b \sqcap a)$. Therefore, in the particular case that $a = b$, from the idempotency of $\sqcap$, we would obtain: $\underline{0} = a \,\Delta\, a$. ■

**Proposition 4** *Any Boolean ring $(L, \Delta, \sqcap)$ is a commutative ring.*

PROOF. It follows from the equality

$$\underline{0} = (a \sqcap b) \,\Delta (b \sqcap a)$$

in the previous proof and the facts that $(L, \Delta)$ is a group (and therefore the opposite is unique) and that any element is its own opposite. ■

**Proposition 5** *A Boolean ring $(L, \Delta, \sqcap)$ can be defined from a Boolean algebra $(L, \sqcup, \sqcap, \,')$: for all elements $a$, $b \in L$:*

$$a \Delta b = (a \sqcap b') \sqcup (a' \sqcap b)$$

*Conversely, a Boolean algebra $(L, \sqcup, \sqcap, \,')$ can be defined from a Boolean ring $(L, \Delta, \sqcap)$: for all elements $a$, $b \in L$:*

$$a \sqcup b = (a \triangle b) \triangle (a \sqcap b)$$

*and if $\underline{0}$ and $\underline{1}$ are the neutral elements of $\Delta$ and $\sqcap$ (respectively), then $\underline{0}$ is the least, $\underline{1}$ is the greatest, and:*

$$a' = a \,\Delta\, \underline{1}$$

**Example 6** $(\mathcal{P}(E), \cup, \cap, \,')$ *is a Boolean algebra (least: $\emptyset$; greatest: $E$). If we define the* symmetric difference *of two elements of $\mathcal{P}(E)$, $A$ and $B$, as:*

$$A \triangle B = (A \cap B') \cup (A' \cap B)$$

*we have that $(\mathcal{P}(E), \triangle, \cap)$ is a commutative ring with unit: all properties are a straightforward consequence of the same property of the Boolean algebra or can be obtained from a simple symbolic manipulation (like the commutativity and associativity of $\triangle$). Moreover:*

- *$\emptyset$ is the neutral element for $\triangle$ (and the absorbent element of $\cap$),*
- *$E$ is the neutral element for $\cap$,*
- *$\forall A \in \mathcal{P}(E)$, $A \triangle A = \emptyset$, so any element in $\mathcal{P}(E)$ is its own opposite.*

**Example 7** $(\mathcal{P}(E), \triangle, \cap)$ *is a Boolean ring. If we define in this ring the following operation:*

$$A \sqcup B = (A \triangle B) \triangle (A \cap B)$$

*the operation $\cup$ is obtained. If we define*

$$A' = A \,\Delta\, E$$

*the usual complement of a set is obtained.*

## 3 A polynomial model for propositional logic

We can obtain a polynomial model for the Boolean algebra associated to (propositional) Boolean Logic, as suggested in Figure 1.

$$\begin{tikzcd}
\boxed{\text{propositional Boolean algebra}} \arrow[r, leftrightarrow, "\textit{Boolean algebra}", "\textit{isomorphism}"'] \arrow[d, leftrightarrow] & \boxed{\text{polynomial Boolean algebra}} \arrow[d, leftrightarrow] \\
\boxed{\text{propositional Boolean ring}} \arrow[r, leftrightarrow, "\textit{ring}", "\textit{isomorphism}"'] & \boxed{\text{polynomial (Boolean) ring}}
\end{tikzcd}$$

Figure 1. Isomorphisms between polynomial and propositional Boolean algebras and Boolean rings.

In logic, we usually work in the structure in the upper left corner of the diagram of Figure 1. Meanwhile, in commutative algebra we normally work in the structure in the lower right corner of this diagram. But we can move from the propositional Boolean algebra to the polynomial ring following the two alternative paths in this diagram. The advantages of working in a polynomial ring are that:

- there is a powerful tool that solves the *ideal membership problem*: Gröbner bases,
- there are many implementations of Buchberger's algorithm for computing Gröbner bases, i.e., performing effective calculations in algebra. In fact, most CAS, like *Maple*, *Mathematica*, *Axiom*, *MuPAD*, *Reduce*, *Derive*, *CoCoA*, *Singular*, *Risa/Asir*,... include implementations of this algorithm.

As a consequence, we shall be able to implement very easily an algebraic tool to perform effective computations in propositional Boolean logic. Moreover, it will be possible to extend this tool to perform effective computations in modal multi-valued logics (like Łukasiewicz, Kleene and Bochvar). Finally, it will be possible to use the same computational tool to perform verification and knowledge extraction in RBES based on both classic Boolean and modal multi-valued logics.

### 3.1 Building the Taylor-made polynomial (Boolean) ring $(\mathcal{A}, +, \cdot)$

Let us build a polynomial Boolean ring $(\mathcal{A}, +, \cdot)$ so that:

- every element is its own opposite, i.e., for all $a \in \mathcal{A}$, $a + a = 2 \cdot a = 0$, so it would be reasonable to consider $\mathcal{A}$ as a ring over a field with characteristic 2,
- idempotency holds for the second operation, so it would be reasonable to consider the residue class ring over the polynomial ideal $\langle p^2 - p, q^2 - q, \ldots, r^2 - r\rangle$, where $p$, $q$, $\ldots$, $r$ are the polynomial variables.

Therefore, we shall consider:

$$\mathcal{A} = \mathbb{Z}_2[p, q, \ldots, r]/\langle p^2 - p, q^2 - q, \ldots, r^2 - r\rangle$$

**Proposition 6** *The polynomial product in $\mathcal{A}$ is idempotent.*

PROOF. It is enough to take into account that any polynomial $a \in \mathcal{A}$ can be written:

$$a = \delta_0 + \delta_p \cdot p + \delta_q \cdot q + \cdots + \delta_r \cdot r + \delta_{pq} \cdot p \cdot q + \cdots + \delta_{pqr} \cdot p \cdot q \cdot r + \cdots$$

where all the $\delta$ belong to $\mathbb{Z}_2$, and compute $a^2$. ■

It is well-known that an algebraic structure like $(\mathcal{A}, +, \cdot)$ is a ring (denoted *residue class ring*) and, therefore, we have the following:

**Corollary 1** *The ring $(\mathcal{A}, +, \cdot)$ is a Boolean ring.*

**Proposition 7** *All elements in $(\mathcal{A}, +, \cdot)$ are zero divisors (in fact an analogous result holds in any Boolean ring).*

PROOF. $a \cdot (1 + a) = a + a^2 = a + a = 2 \cdot a = 0.$ ■

## 3.2 The polynomial Boolean algebra $(\mathcal{A}, \tilde{+}, \cdot, 1+)$ corresponding to the polynomial (Boolean) ring $(\mathcal{A}, +, \cdot)$

Let us define a (polynomial) Boolean algebra $(\mathcal{A}, \sqcup, \cdot, ')$ from the (polynomial) Boolean ring $(\mathcal{A}, \tilde{+}, \cdot)$, as shown in Proposition 5. For all $a, b \in \mathcal{A}$:

$$a \sqcup b = (a + b) + (a \cdot b)$$
$$a' = a + 1$$

The first operation will be hereinafter denoted $\tilde{+}$ and $1+$ will denote the addition of $1$. $0$ is the least and $1$ is the greatest of the Boolean algebra $(\mathcal{A}, \tilde{+}, \cdot, 1+)$: for all $a, b \in \mathcal{A}$,

$$a\tilde{+}0 = a + 0 + a \cdot 0 = a \qquad a \cdot 0 = 0$$
$$a\tilde{+}1 = a + 1 + a \cdot 1 = 1 \qquad a \cdot 1 = a$$

Unlike what happens in a ring, it is not usual to establish priorities for the operations in a Boolean algebra (as the structure is completely "symmetrical"). Nevertheless, we shall consider, hereinafter, that $\cdot$ is prioritary w.r.t. $\tilde{+}$.

**Proposition 8** *The lattice order defined in $(\mathcal{A}, \tilde{+}, \cdot, 1+)$ as suggested in Proposition 2: $\forall a, b \in \mathcal{A}$, $a \leq b \Leftrightarrow a \cdot b = a$ is, precisely, "is a multiple" ($a$ is a multiple of $b \Leftrightarrow \exists k \in \mathcal{A} : a = b \cdot k$).*

PROOF. $\Rightarrow)$ $\forall a, b \in \mathcal{A}$: $a \leq b \Leftrightarrow\ a \cdot b = a\ \Rightarrow\ a$ is a multiple of $b$
$\Leftarrow)$ $\forall a, b \in \mathcal{A}$: $a$ is a multiple of $b \Leftrightarrow \exists k \in A : a = b \cdot k\ \Rightarrow a \cdot b = (b \cdot k) \cdot b = b^2 \cdot k = b \cdot k = a \Leftrightarrow a \leq b$. ■

**Proposition 9** *For all $a, b \in \mathcal{A}$:*

(1) $a \cdot b = a$

(2) $a\tilde{+}b = b$

(3) $(1 + a)\tilde{+}b = 1$

(4) $a \cdot (1 + b) = 0$

*are equivalent (an analogous result holds in any Boolean algebra).*

# 4 The Boolean algebra isomorphism $\varphi$

Let $\vee$, $\wedge$, $\neg$, $\rightarrow$ denote the logic disjunction, conjunction, negation and implication, respectively.

Let $(\mathcal{C}, \vee, \wedge, \neg, \rightarrow)$ be the Boolean algebra of the propositions that can be constructed using a finite number of propositional variables $P, Q, \ldots, R$. Let us denote tautology by $\underline{1}$ and contradiction by $\underline{0}$.

Let us consider the Boolean algebra $(\mathcal{A}, \tilde{+}, \cdot, 1+,$ "is a multiple"$)$, where $\mathcal{A}$ is the residue class ring

$$\mathcal{A} = \mathbb{Z}_2[p, q, \ldots, r]/\langle p^2 - p, q^2 - q, \ldots, r^2 - r\rangle$$

We define:

$$\varphi\colon (\mathcal{C}, \vee, \wedge, \neg, \rightarrow) \longrightarrow (\mathcal{A}, \tilde{+}, \cdot, 1+, \text{“is a multiple”})$$

in the following way. For propositional variables

$$\begin{aligned} P &\longrightarrow p \\ Q &\longrightarrow q \\ &\cdots\cdots \\ R &\longrightarrow r \end{aligned}$$

and for any $A$, $B \in \mathcal{C}$

$$\begin{aligned} A \vee B &\longrightarrow a \tilde{+} b \\ \neg A &\longrightarrow 1 + a \end{aligned}$$

Therefore, as an immediate consequence of the De Morgan laws:

$$A \wedge B \longrightarrow a \cdot b$$

Moreover, as the (lattice) order can be obtained from the operations of the lattice, $\varphi$ preserves the ordering.

**Proposition 10** *$\varphi$ is well defined.*

PROOF. For all propositions $A$, $B$:

$$\begin{gathered} A \leftrightarrow B \Rightarrow A \rightarrow B \text{ and } B \rightarrow A \Rightarrow \\ \Rightarrow \varphi(A) \text{ is a multiple of } \varphi(B) \text{ and } \varphi(B) \text{ is a multiple of } \varphi(A) \Leftrightarrow \\ \Leftrightarrow a \text{ is a multiple of } b \text{ and } b \text{ is a multiple of } a \Leftrightarrow a = b. \quad \blacksquare \end{gathered}$$

**Corollary 2** *$\varphi$ is an order-preserving homomorphism, and $\varphi(\underline{1}) = 1, \varphi(\underline{0}) = 0$.*

**Remark 1** *In order to save space, we shall sometimes write $b|a$ (“$b$ divides $a$”) instead of “$a$ is a multiple of $b$”.*

**Proposition 11** *$\varphi$ is surjective.*

PROOF. The elements of the (Boolean) ring $\mathcal{A}$ (they are the same as those of the Boolean algebra $\mathcal{A}$) are a linear algebraic combination of the polinomial variables, and:

$$\begin{gathered} \varphi(P \wedge Q) = p \cdot q \\ \varphi((\neg P \wedge Q) \vee (P \wedge \neg Q)) = p + q. \quad \blacksquare \end{gathered}$$

**Proposition 12** *$\varphi$ is injective.*

PROOF. Let us suppose that for two propositions, $A$ and $B$, we have: $\varphi(A) = a$, $\varphi(B) = b$ and $a = b$. As $a = b$, we have: $a|b$ and $b|a$. But $\varphi$ preserves the ordering (i.e., $\rightarrow$ and “is a multiple” do correspond), so:

$$b|a \Rightarrow A \rightarrow B; \qquad a|b \Rightarrow B \rightarrow A$$

and therefore $A \leftrightarrow B$. $\blacksquare$

**Remark 2** *To be precise, we really consider $\mathcal{C}/\leftrightarrow$ and $\mathcal{A}/=$ (Lindenbaum algebras) in order for the relations $\rightarrow$ and $\leq$ to be anti-symmetric.*

**Example 8** *Let $P$ and $Q$ be the propositional variables of $\mathcal{C}$. Then $\mathcal{C}$ has 16 elements and $(\mathcal{C},\rightarrow)$ is the transitive closure of the diagram in Figure 2. It corresponds in $\varphi$ with $(\mathcal{A}$, "$is\ a\ multiple$"), where*

$$\mathcal{A}=\mathbb{Z}_2[p,q]/\langle p^2-p,q^2-q\rangle$$

*and "is a multiple" is the transitive closure of the diagram in Figure 3.*

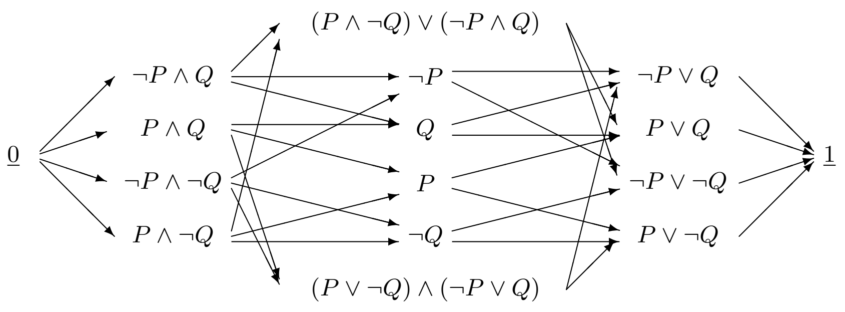


Figure 2. $(\mathcal{C},\rightarrow)$ when there are two propositional variables.

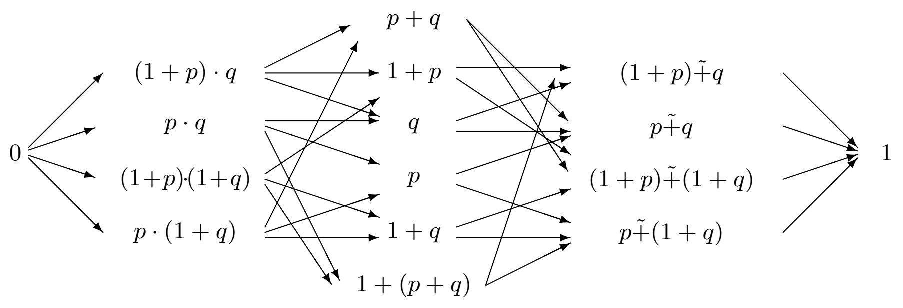


Figure 3. $(\mathcal{A},$"is a multiple") when there are two polynomial variables.

## 4.1 Ideals of a Boolean algebra and ideals of a ring

**Definition 8** *Let $(\mathcal{R},+,\cdot)$ be a commutative ring. A subset $I\subseteq\mathcal{R}$ is said to be an* ideal *of $\mathcal{R}$ if and only if:*

1) $\forall i,i'\in I: i+i'\in I$

2) $\forall i\in I,\forall a\in\mathcal{R}: a\cdot i\in I$

*(i.e. $I$ is a subring of $\mathcal{A}$ such that the product of an element of the ring by an element of the subring always belongs to the subring).*

**Definition 9** *Let $(\mathcal{R},+,\cdot)$ be a commutative ring. Let $S \neq \emptyset$, $S \subseteq \mathcal{R}$. The* ideal generated by *$S = \{p_1,\dots,p_m\}$, denoted $\langle p_1,\dots,p_m\rangle$, is the smallest ideal containing S. If $S=\{q\}$, the ideal generated by $q$, $\langle q\rangle$, is said to be a* principal ideal*, and results to be:*

$$\langle q\rangle = \{\, a \in \mathcal{A} : q|a \,\}$$

Despite the concept of *ideal* is usually associated with rings, ideals are also defined in Boolean algebras. In such case, the definition is different (based on the lattice order), and only ideals generated by a single element are considered.

**Definition 10** *In the propositional Boolean algebra $(\mathcal{C},\vee,\wedge,\neg,\rightarrow)$, the (principal)* ideal generated by *$Q$ is defined as*

$$E_Q = \{\, X \in \mathcal{C} : X \rightarrow Q \,\}$$

*Similarly, in the polynomial Boolean algebra $(\mathcal{A},\tilde{+},\cdot,1+,\text{"is a multiple"})$, the (principal) ideal generated by $q \in \mathcal{A}$ is:*

$$E_q = \{\, a \in \mathcal{A} : a \text{ is a multiple of } q \,\}$$

**Proposition 13** *As $\varphi$ preserves the ordering, the ideals of the Boolean algebra $(\mathcal{C},\vee,\wedge,\neg,\rightarrow)$ do correspond in $\varphi$ with the ideals of the Boolean algebra $(\mathcal{A},\tilde{+},\cdot,1+,\text{"is a multiple"})$.*

**Proposition 14** *The ideals of the Boolean algebra $(\mathcal{A},\tilde{+},\cdot,\text{"is a multiple"})$ are obviously ideals of the ring $(\mathcal{A},+,\cdot,\text{"is a multiple"})$.*

**Theorem 1** *$(\mathcal{A},+,\cdot)$ is a ring where all ideals are principal ones.*

PROOF. Let $s_1, s_2, \dots, s_n \in \mathcal{A}$. We shall prove that $\langle s_1,s_2,\dots,s_n\rangle = \langle s_1\tilde{+}s_2\tilde{+}\cdots\tilde{+}s_n\rangle$ in two steps:

i) $\tilde{+}$ is an internal operation in the ideal (because $+$ and $\cdot$ also are internal operations) and therefore:

$$s_1\tilde{+}s_2\tilde{+}\cdots\tilde{+}s_n \in \langle s_1,s_2,\dots,s_n\rangle \Rightarrow \langle s_1\tilde{+}s_2\tilde{+}\cdots\tilde{+}s_n\rangle \subseteq \langle s_1,s_2,\dots,s_n\rangle$$

ii) That $s_1\tilde{+}s_2\tilde{+}\cdots\tilde{+}s_n|s_i$; $i=1,\dots,n$ can be easily proven. But, then: $s_i \in \langle s_1\tilde{+}s_2\tilde{+}\cdots\tilde{+}s_n\rangle$; $i=1,\dots,n$ and, consequently, the minimum ideal that contains $\{s_1,s_2,\dots,s_n\}$, is contained in $\langle s_1\tilde{+}s_2\tilde{+}\cdots\tilde{+}s_n\rangle$, i.e.:

$$\langle s_1,s_2,\dots,s_n\rangle \subseteq \langle s_1\tilde{+}s_2\tilde{+}\dots\tilde{+}s_n\rangle. \quad \blacksquare$$

**Remark 3** *Let us observe that $(\mathcal{A},+,\cdot)$ is not a "Principal Ideals Domain" because it is not an integrity domain (see Proposition 7).*

**Corollary 3** *As a consequence of Proposition 13, Proposition 14 and Theorem 1, the ideals of the polynomial Boolean algebra $(\mathcal{A},\tilde{+},\cdot,\text{"is a multiple"})$ are precisely the ideals of the polynomial ring $(\mathcal{A},+,\cdot)$.*

## 5 Introductory note about multi-valued propositional logics

A simple introduction to the topic can be found in [37].

## 5.1 Kleene's and Łukasiewicz's three-valued logic

Kleene's and Łukasiewicz's multi-valued logics are two of the most commonly used multi-valued logics. We shall begin introducing these three-valued logics, in which a third truth value is considered.

In Kleene's logic this third valued is "undecided". If an assertion is assigned the truth value "undecided" that means that at present, we can't assign a truth-value to this conjecture (but the conjecture is either true or false). For instance, if we are reasoning in a medical diagnostic RBES, until we receive the result of a test, its truth value is "undecided", but it will become either true or false in the future (when we receive the results). Kleene's $A \to B$ is equivalent to $\neg A \vee B$.

Two modal unary connectives, necessary ($\Box$) and possible ($\Diamond$) are usually considered in multi-valued logics. Kleene's three valued logic truth-tables are detailed in Tables 1 and 2 (note that $0$ represents "false", $1$ represents "undecided" and $2$ represents "true"). It would also be possible to assign numbers to truth values in a different way. For instance, in Kleene's three valued logic, it is also common to consider that $0$ represents "false", that $2$ represents "undecided" and that $1$ represents "true". In such case the corresponding truth tables would obviously change.

| $\neg$ | |
|---|---|
| 0 | 2 |
| 1 | 1 |
| 2 | 0 |

| $\Box$ | |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 2 |

| $\Diamond$ | |
|---|---|
| 0 | 0 |
| 1 | 2 |
| 2 | 2 |

Table 1. Truth-tables of Kleene's three-valued logic unary connectives.

| $\wedge$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 2 | 0 | 1 | 2 |

| $\vee$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 1 | 2 |
| 1 | 1 | 1 | 2 |
| 2 | 2 | 2 | 2 |

| $\to$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 2 | 2 | 2 |
| 1 | 1 | 1 | 2 |
| 2 | 0 | 1 | 2 |

| $\leftrightarrow$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 2 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 2 |

Table 2. Truth-tables of Kleene's three-valued logic binary connectives.

In Łukasiewicz's logic the third truth value considered is "indeterminate". If a statement is considered to be indeterminate, it is neither true not false, but indeterminate in a metaphysical sense. All connectives have the same truth tables as in Kleene's logic, except $\to$ and $\leftrightarrow$ (see Table 3).

| $\to$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 2 | 2 | 2 |
| 1 | 1 | 2 | 2 |
| 2 | 0 | 1 | 2 |

| $\leftrightarrow$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 2 | 1 | 0 |
| 1 | 1 | 2 | 1 |
| 2 | 0 | 1 | 2 |

Table 3. Truth-tables of the Łukasewicz's three-valued logic binary connectives whose truth-tables are different from Kleene's ones.

**Remark 4** *It is also possible to consider more than three truth values, so that the confidence in the truthfulness of statements can be graded. Moreover, there are other multivalued logics aside from Kleene's and Łukasewicz's.*

### 5.2 Tautological consequence

**Definition 11** *A propositional formula $A_0$ is a* tautological consequence *of the propositional formulae $A_1$, $A_2$, …, $A_m$, denoted $\{A_1, A_2, \dots, A_m\} \models A_0$, if and only if whenever the formulae $A_1$,…,$A_m$ hold (i.e., take the truth value "true"), then $A_0$ also holds (i.e., takes the truth value "true").*

**Remark 5** *Let us observe that:*

- *in Boolean logic, there are formulae that can only take the truth value "false", such as $P \wedge \neg P$,*
- *in multi-valued logic, there are also formulae that can only take the truth value "false", such as $\Box P \wedge \Box\neg P$ .*
- *in multi-valued logic, there are formulae that, although can take other truth values different from "false", they can never take the truth value "true", such as $P \wedge \neg P$ (observe that not only $\Box P \wedge \Box\neg P \models P \wedge \neg P$, but also $P \wedge \neg P \models \Box P \wedge \Box\neg P$).*

### 5.3 The isomorphism $\varphi$ in the multi-valued case

Let $(\mathcal{C}, \vee, \wedge, \neg, \rightarrow)$ be a $p$-valued logic, where $p$ is a prime number (we require $p$ to be a prime in order for $\mathbb{Z}_p$ to be a field) and $\mathcal{C}$ is the set of propositions that can be constructed using the propositional variables $X_1$, $X_2$, …, $X_n$. If in the residue class ring

$$\mathcal{A} = \mathbb{Z}_p[x_1, x_2, \dots, x_n]/\langle x_1^p - x_1, x_2^p - x_2, \dots, x_m^p - x_m\rangle$$

adequate polynomial translations of the logical connectives are defined (so that they behave as suggested by the truth-tables), the natural correspondence $\varphi$, defined as in the Boolean case (Section 4), is also an isomorphism.

## 6 The main Theorem

### 6.1 A brief note about Gröbner bases

In the early 60's, both Heisuke Hironaka and Bruno Buchberger independently proved that, for each polynomial ideal, a basis completely identifying it always exists. They denoted their bases as *standard bases* and *Gröbner bases* (GB) [4, 5], respectively. The latter's great advantage was that it provided a constructive method (Buchberger's algorithm). Some GB are particularly important: we call them *reduced Gröbner bases*. We say that a Gröbner basis is reduced if and only if the leading coefficient of all its polynomials is 1 and we cannot simplify any of its polynomials by adding a linear algebraic combination of the rest of the polynomials in the basis. The input to Buchberger's algorithm is a polynomial set, a term order (for instance, "total degree" or "pure lexicographical"), and a variable order (for instance, $x > y > z$) and its output is the ideal's reduced GB with respect to the specified term and variable orders.

The key point is that, once the term order and the variable order are fixed, such a reduced GB completely characterizes the ideal: any ideal has a unique reduced GB. As a consequence we have two key results:

i) two sets of polynomials generate the same ideal if and only if their reduced GB are the same,

ii) $\{1\}$ is the only reduced GB for the ideal that is equal to the whole ring.

An elementary introduction to the use of GB in algebraic system solving can be found in [33, 34]. For detailed introductions see, for instance, [1, 2, 9, 16, 38]. There are interesting applications in other fields like transportation engineering [32, 30].

### 6.2 Connecting "To be a tautological consequence" with an ideal membership

The main theorem is the following:

**Theorem 2** *A logical formula, $A_0$, is a tautological consequence of the set of formulae $\{A_1, A_2, \ldots, A_m\}$ in a certain $p$-valued modal logic (where $p$ is a prime number) if and only if the polynomial translation of $\neg A_0$ in $\mathcal{A} = \mathbb{Z}_p[x_1, x_2, \ldots, x_n]/\langle x_1^p - x_1, x_2^p - x_2, \ldots, x_m^p - x_m\rangle$, $\varphi(\neg A_0)$, belongs to the ideal $\langle \varphi(\neg A_1), \varphi(\neg A_2), \ldots, \varphi(\neg A_m)\rangle$ of $\mathcal{A}$.*

The key point is that this theorem can take advantage of the two important results stated at the end of the previous section. The proof of this theorem is really long and can be found in [19, 27].

## 7 Some introductory notes about RBES

A RBES consists of an "input", an "output", a "knowledge base" and an "inference engine". A graphic user interface is frequently provided. The *input* of a RBES is concerned with the information related to the environment of the RBES. Since the environment may change, this information is also subject to change as times goes by. This information is described by means of a finite number of different atomic propositions: such propositions are termed as *input atomic propositions*. Input atomic propositions determine what we will call potential facts (see Definition 13). The *output* of a RBES is concerned with the information inferred by the RBES. It is described by means of a finite number of atomic propositions which we shall call *output atomic propositions*. The *knowledge base* (*KB*) of the RBES is concerned with the information contained in the system, which is used along with the input of the RBES to infer the output of the system. In a RBES based on propositional Boolean logic, this knowledge-base will be mostly represented by a finite set of rules.

**Remark 6** *We shall use hereinafter the following notation: $\maltese$ may mean:*

- *no symbol at all or "negation" (denoted $\neg$), if the underlying logic is classic Boolean,*
- *no symbol at all, "negation", "possibility" (denoted $\Diamond$), "necessity" (denoted $\Box$), or a combination of these symbols, like "necessarily-not" (denoted $\Box\neg$), equivalent to $\neg\Diamond$, if the underlying logic is a modal multi-valued one.*

### 7.1 Rules

**Definition 12** *The* KB *consists of a certain number of logical formulae of the form:*

$$\wedge_{i=1}^{k} \maltese X[i] \longrightarrow \vee_{j=1}^{l} \maltese X[j]$$

*known as* production rules *(and usually refereed to as, simply,* rules*).*

For example, formula

$$X[1] \wedge X[2] \wedge \neg X[3] \wedge \Diamond X[4] \wedge \neg X[5] \longrightarrow X[13] \vee \Box\neg X[17]$$

is read as: "IF $X[1]$ and $X[2]$ and *not* $X[3]$ and possibly $X[4]$ and *not* $X[5]$ HOLD, THEN $X[13]$ or necessarily-not $X[17]$ HOLDS".

Let us underline that there is no constraint in the structure of the rules: the ocurrences of "$\wedge$" in the consequent and of "$\vee$" in the antecedent of a rule can be avoided (if such symbols ocurred in a rule, the rule could be split into rules without these symbols).

## 7.2 Potential facts and facts. Integrity constraints

**Definition 13** *A formula $A$ is a* potential fact *if and only if $A \equiv \maltese X[i]$, where $X[i]$ is an input atomic proposition.*

**Definition 14** *That a given set of facts (i.e., a subset of the set of potential facts that is stated as true) is* consistent *means that, for each propositional variable appearing in the potential facts of the system, $X[k]$, at most one expression of the form $\maltese X[k]$, is chosen. That a given set of facts is* maximal *means that, for each propositional variable appearing in the potential facts of the system, $X[k]$, exactly one expression of the form $\maltese X[k]$, is chosen.*

**Definition 15** *An* integrity constraint *(*IC*) is a piece of knowledge added by the experts and expressing that some potential facts cannot hold at the same time.*

For instance an IC could be: *man* $\wedge$ *pregnant*. The negation of the ICs (sometimes denoted "NICs") have to be added to the KB.

## 7.3 The inference engine. RBES inconsistency

The *inference engine* (*IE*) is an automated tool (in our case, a program in the CAS), that verifies consistency and draws tautological consequences from the information contained in the KB.

**Definition 16** *That a formula $A$ be obtained by* forward firing *of the formulae in the union, $\mathcal{G}$, of a consistent set of facts and the set of all production rules and the set of the negation of the integrity constraints of a RBES means that $\mathcal{G} \models A$.*

**Definition 17** *A RBES is said to be* inconsistent *if there is a consistent set of facts verifying that contradiction can be obtained by forward firing of these facts and the rules and the negation of the integrity constraints in the KB. Otherwise the RBES is said to be* consistent*.*

# 8 Knowledge extraction in RBES through the use of an algebraic inference engine

This extension of the algebraic method to RBES was already mentioned in [18] and detailed in [26].

## 8.1 A polynomial model for the propositional Boolean algebra associated to RBES

The Boolean algebra associated to a RBES whose underlying logic is classic Boolean logic is a structure $(\mathcal{C}^*, \vee, \wedge, \neg, \rightarrow)$ where $\rightarrow$ is the relation obtained applying the rules of logical deduction to the implications of $\mathcal{C}$ and the rules, facts and the negations of the integrity constraints of the RBES (consequently, $\rightarrow$ is not the usual implication), and $\mathcal{C}^*$ is the set of equivalence classes defined by this enlarged equivalence relation $\leftrightarrow$ in $\mathcal{C}$.

If the rules $Rule_1, \ldots, Rule_v$, the facts $Fact_1, \ldots, Fact_m$ and the integrity constraints $IC_1, \ldots, IC_u$ of a RBES are added as true to the Boolean algebra $\mathcal{C}$ of the previous section, the structure obtained is isomorphic to the image of $\mathcal{A}$ in the natural surjective homomorphism

$$\psi \colon \mathcal{A} \longrightarrow \mathcal{A}/J$$

where $J$ is the ideal

$$\langle \varphi(\neg Rule_1), \ldots, \varphi(\neg Rule_v), \varphi(\neg Fact_1), \ldots, \varphi(\neg Fact_m), \varphi(\neg IC_1), \ldots, \varphi(\neg IC_u) \rangle$$

### 8.2 The main theorem adapted to knowledge extraction in RBES

The main theorem detailed above can be adapted to treat the logic underlying a RBES:

**Theorem 3** *A formula $A_0$ can be obtained by forward firing of the formulae in the union of a consistent set of facts and the set of all production rules and the negation of the integrity constraints of a* RBES*, if and only if, the polynomial translation of the negation of $A_0$ belongs to the ideal $J$ of $\mathcal{A} = \mathbb{Z}_p[x_1, x_2, \ldots, x_n]/\langle x_1^p - x_1, x_2^p - x_2, \ldots, x_n^p - x_n\rangle$, generated by the polynomial translations of the negations of the given potential facts, of the negations of all the production rules, and the integrity constraints of the* RBES.

**Remark 7** *Observe that, denoting $I = \langle x_1^p - x_1, x_2^p - x_2, \ldots, x_m^p - x_n\rangle$, we could alternatively check that $A_0$ belongs to the ideal $I + J$ of $\mathbb{Z}_p[x_1, x_2, \ldots, x_n]$. We usually work this way as most* CAS *are not prepared to perform computations in a residue class ring.*

We have extensively used this approach in RBES design and verification. [20] is an interesting example in medicine (regarding techniques for coronary surgery), where we could find a situation that could lead to an inconsistency and was not taken into account by the panel of experts.

## 9 RBES strong and weak inconsistency. Algebraic interpretation

An introduction to verification can be found in [17]. This section summarizes [31].

We shall make the following distinction in the multi-valued case:

**Definition 18** *We shall say that there is* strong inconsistency *if and only if the conjunction of the facts in a certain consistent set of facts, the rules and the negations of integrity constraints of the* RBES*, can only take the truth value "false".*

**Definition 19** *We shall say that there is* weak inconsistency *if and only if all formulae can be obtained by forward firing of the formulae in the union of a certain consistent set of facts, the set of all production rules and the set of the negations of integrity constraints of the* RBES.

**Proposition 15** *Independently of the number of truth values ($p$) of the underlying logic:*

$$\text{strong inconsistency} \Rightarrow \text{weak inconsistency.}$$

PROOF. If there is strong inconsistency, we have a consistent set of facts $\{Fact_1, \ldots, Fact_m\}$ such that:

$$Fact_1 \wedge \ldots \wedge Fact_m \wedge Rule_1 \wedge \ldots \wedge Rule_v \wedge NIC_1 \wedge \ldots \wedge NIC_u$$

can only take the truth values false. But

$$\begin{aligned}\{Fact_1, \ldots, &Fact_m, Rule_1, \ldots, Rule_v, NIC_1, \ldots, NIC_u\} \\ &\models Fact_1 \wedge \ldots \wedge Fact_m \wedge Rule_1 \wedge \ldots \wedge Rule_v \wedge NIC_1 \wedge \ldots \wedge NIC_u\end{aligned}$$

so a contradiction can be obtained by forward firing of the formulae in

$$\{Fact_1, \ldots, Fact_m, Rule_1, \ldots, Rule_v, IC_1, \ldots, IC_u\},$$

and, therefore, we have weak inconsistency. ■

**Proposition 16** *If the underlying logic is Boolean:*

$$\text{weak inconsistency} \Rightarrow \text{strong inconsistency.}$$

PROOF. If there is weak inconsistency, we have a consistent set of facts $\{Fact_1, \ldots, Fact_m\}$ such that any formula follows of the formulae in $\{Fact_1, \ldots, Fact_m, Rule_1, \ldots, Rule_v, NIC_1, \ldots, NIC_u\}$ by forward firing. In particular,

$$\{Fact_1, \ldots, Fact_m, Rule_1, \ldots, Rule_v, NIC_1, \ldots, NIC_u\} \models \text{contradiction}$$

But we are considering that the underlying logic is Boolean, so:

$$Fact_1 \wedge \ldots \wedge Fact_m \wedge Rule_1 \wedge \ldots \wedge Rule_v \wedge NIC_1 \wedge \ldots \wedge NIC_u \rightarrow \text{contradiction}$$

and therefore $Fact_1 \wedge \ldots \wedge Fact_m \wedge Rule_1 \wedge \ldots \wedge Rule_v \wedge NIC_1 \wedge \ldots \wedge NIC_u$ can only take the truth value "false". ■

**Remark 8** *The previous result does not hold if the underlying logic is a multi-valued one.*

**Example 9** *Let us consider a small RBES whose KB consists of the rules:*

$$R1 : X[1] \rightarrow X[3]$$
$$R2 : X[2] \rightarrow \neg X[3]$$

*and whose underlying logic is Kleene's three-valued logic. The potential facts are $X[1]$ and $X[2]$. If we draw consequences from the maximal set of potential facts $\{X[1], X[2]\}$, we obtain $X[3] \wedge \neg X[3]$. As this formula can only take the truth values "undecided" or "false", any formula can be obtained by forward firing of it, and, consequently, by forward firing of the formulae in $\{X[1], X[2], R1, R2\}$. Therefore we have weak inconsistency. Meanwhile, we cannot obtain a formula that is always "false" by forward firing of the conjunction of the facts in any consistent set of facts and $R1 \wedge R2$. Therefore, we do not have strong inconsistency.*

**Corollary 4** *There is weak inconsistency if and only if*

$$1 \in \langle f_{\neg}(\varphi(Rules)), f_{\neg}(\varphi(Facts)), f_{\neg}(\varphi(NICs))\rangle.$$

But, what is the meaning of strong inconsistency in the polynomial model?

**Definition 20** *The algebraic variety corresponding to the polynomial ideal $I$ is the set of points of the respective affine space which satisfy all polynomials in the ideal $I$. We shall denote it by $V(I)$.*

**Lemma 1** *For any positive prime integer $p > 1$, we have:*

i) *There are non-zero ideals in $\mathbb{Z}_p[x_1, x_2, \ldots, x_m]$ whose algebraic variety is the whole affine space.*

ii) *The only ideal in $\mathcal{A} = \mathbb{Z}_p[x_1, x_2, \ldots, x_m]/I;\ I = \langle x_1^p - x_1, x_2^p - x_2, \ldots, x_m^p - x_m\rangle$ whose variety is the whole affine space is the zero ideal.*

PROOF.

i) For example $x \cdot (x-1) = x^2 - x \in \mathbb{Z}_2[x]$ is not the null polynomial but its value is always $0$ in $\{0, 1\}$. Therefore $\langle x^2 + x\rangle$ verifies condition i). The same occurs with $x \cdot (x-1) \cdot (x-2) \in \mathbb{Z}_3[x]$. Its value in $\{0, 1, 2\}$ is always $0$. This construction can be obviously generalized to $\mathbb{Z}_p[x]$.

ii) Case I: Univariate polynomials. If $0, 1, \ldots, p-1$ are roots of a polynomial in $\mathcal{A} = \mathbb{Z}_p[x]/\langle x^p - x\rangle$, then the polynomial is either $0$ or it can be factored as

$$x \cdot (x-1) \cdot \cdots \cdot (x-(p-1)) \cdot \cdots$$

Therefore, if this polynomial is different from $0$, its total degree is greater or equal than $p$. But in $\mathcal{A}$ all polynomials are simplified to polynomials of degree lesser than $p$, so this polynomial simplifies to $0$.
Case II: Multi-variate polynomials. It can be proved by applying the previous case to the intersections of the hypersurface by hyperplanes parallel to the "vertical" axis. ■

**Theorem 4** *There is strong inconsistency if and only if*

$$V(\langle\varphi(\mathit{Rules} \wedge \mathit{Facts} \wedge \mathit{NICs})\rangle)$$

*is the whole (respective) affine space.*

PROOF.

$$\mathit{Rules} \wedge \mathit{Facts} \wedge \mathit{NICs} \text{ can only take the truth value “false”} \Leftrightarrow$$
$$\Leftrightarrow \text{polynomial } \varphi(\mathit{Rules} \wedge \mathit{Facts} \wedge \mathit{NICs}) \text{ can only take the value } 0 \Leftrightarrow$$
$$\Leftrightarrow V(\langle\varphi(\mathit{Rules} \wedge \mathit{Facts} \wedge \mathit{NICs})\rangle) \text{ is the whole (respective) affine space.} \quad ■$$

But we can express the previous result in terms of ideals instead of algebraic varieties, what makes it easily decidable using Gröbner bases:

**Corollary 5** *There is strong inconsistency if and only if* $\langle\varphi(\mathit{Rules} \wedge \mathit{Facts} \wedge \mathit{NICs})\rangle$ *is the zero ideal of* $\mathcal{A} = \mathbb{Z}_p[x_1, x_2, \ldots, x_m]/I$.

## 10 Backward reasoning in rule based expert systems whose underlying logic is classic Boolean logic

Details on the topic of this section, our most recent result, can be found in [25]. Based on the classical works of automatic discovery of geometric theorems [14, 23] (a topic already advanced in [8]), we have now adapted this idea to a specific task in RBES whose underlying logic is classic Boolean logic: given a set of facts and a goal, finding a new fact such that if it also held, this goal would be inferred. The main result proven in [25] (Theorem 5 below) shows how it is enough to compute one single reduced GB in order to detect them:

**Theorem 5** *Let* $A_1, \ldots, A_k, B \in \mathcal{C}^*$ *be formulae such that* $\{A_1, \ldots, A_k\} \not\models B$. *Let* $C$ *be a potential fact. Let* $G$ *be the reduced* GB *of the ideal* $\langle\varphi(\neg A_1), \ldots, \varphi(\neg A_k), \varphi(B)\rangle + I$. *We have that:*

$$\{A_1, \ldots, A_k, C\} \models B \Leftrightarrow \varphi(C) \in G$$

Observe that the membership relation in the right hand side of the equivalence refers to a membership to the set of polynomials in the reduced GB, not to the ideal generated by that basis.

## 11 Implementation in a Computer Algebra System

As most CAS include an implementation of GB and of "normal forms" (reductions modulo an ideal), to develop an inference engine in a CAS following the method detailed above (based on checking polynomial ideal memberships) is straightforward and surprisingly brief.

### 11.1 Polynomial translation of the logical connectives

For instance, the polynomial expressions corresponding to the basic logical formulae in Łukasiewicz's three-valued logic (if 2 is assigned to "true", 1 to "undetermined" and 0 to "false") are detailed afterwards.

- $\neg M$ is translated into the polynomial: $2 - m$
- $\Diamond M$ is translated into the polynomial: $2m^2$
- $\Box M$ is translated into the polynomial: $m^2 + 2m$
- $M \vee N$ is translated into the polynomial: $m^2n^2 + m^2n + mn^2 + 2mn + m + n$
- $M \wedge N$ is translated into the polynomial: $2m^2n^2 + 2m^2n + 2mn^2 + mn$
- $M \to N$ is translated into the polynomial: $2m^2n^2 + 2m^2n + 2mn^2 + mn + 2m + 2$
- $M \leftrightarrow N$ is translated into the polynomial: $m^2n^2 + m^2n + mn^2 + 2mn + 2m + 2n + 2$

(note that the coefficients of these polynomials belong to $\mathbb{Z}_3$).

The polynomials corresponding to the different logics can be obtained by solving an algebraic system that is obtained from the truth tables.

### 11.2 CoCoA implementation

These polynomial translations can be input almost directly in *CoCoA 4.3* [6, 35], which provides a "Normal Form" command, `NF(pol,I)`, that returns the reduction of polynomial $pol$ modulo ideal $I$.

We have to define the polynomial ring and ask *CoCoA* to use it:

```
USE Z/(3)[x[1..15]];
```

and then we can define the ideal `MEMORY.I`, generated by the expressions $x_i^3 - x_i$ (denoting it this way, it is a global variable, and it can be an input to `NF`):

```
MEMORY.I:=Ideal([x[K_]^3-x[K_] | K_ In 1..15]);
```

and introduce the polynomial expressions for the three-valued connectives of Łukasiewicz modal logic (the operators are prefix ones)[1]:

```
NEG(M):=NF(2-M,MEMORY.I);
POS(M):=NF(2*M^2,MEMORY.I);
NEC(M):=NF(M^2+2*M,MEMORY.I);
OR(M,N):=NF(M^2*N^2+M^2*N+M*N^2+2*M*N+M+N,MEMORY.I);
AND(M,N):=NF(2*M^2*N^2+2*M^2*N+2*M*N^2+M*N,MEMORY.I);
IMP(M,N):=NF(2*M^2*N^2+2*M^2*N+2*M*N^2+M*N+2*M+2,MEMORY.I);
IFF(M,N):=NF(M^2*N^2+M^2*N+M*N^2+2*M*N+2*M+2*N+2,MEMORY.I);
```

For instance, the *CoCoA* implementation of the polynomial expressions of a 7-valued modal logic can be found in [27].

Whether $A_0$ is a consequence of a set of formulae or not, can be checked with *CoCoA* just typing:

```
NF(NEG(A[0]),MEMORY.I+J);
```

(where `J` is the polynomial ideal generated by the polynomial expressions of the negation of the formulae in the given set). If the output of the command is 0, the answer is "yes", otherwise the answer is "no".

[1] In *CoCoA 4.7* these operators would be defined using `Define...EndDefine` instead of directly `:=`.

It can also be directly checked using the Boolean command `IsIn`, that tests ideal memberships, by typing:

```
NEG(A[0]) IsIn MEMORY.I+J;
```

Whether the RBES is inconsistent or not can also be checked using Gröbner bases by typing in *CoCoA*:

```
GBasis(MEMORY.I+J);
```

If the output is 1, the RBES is inconsistent; otherwise (the output is normally a large set of polynomials), that set of facts doesn't lead to an inconsistency.

Using `IsIn` makes it even simpler, as it directly returns "true" or "false", just typing:

```
1 IsIn MEMORY.I+J;
```

## 11.3 Maple implementation

Let us implement the model in *Maple*. We have developed different implementations in *Maple*, but the one presented afterwards is the most modern and fastest [29]. In *Maple* (version $\geq$ 10) it is possible to define a polynomial ring over a finite field using *Maple*'s `Ore_Algebra`. In this way, the `NormalForm` command is computed in such ring. We load the `Gröbner` and `Ore_algebra` packages first, and then define the list of variables, the polynomial ring and the order that will be used by the GB-related commands:

```
with(Groebner):
with(Ore_algebra):
SV:=x[1],x[2],x[3]:
A:=poly_algebra(SV,characteristic=3):
Orde:=MonomialOrder(A,'plex'(SV)):
```

and the ideal $I$ (denoted `iI`, as `I` is a reserved word in *Maple*), using `map` in order to save time:

```
fu:=v->v^3-v:
iI:=map(fu,[SV]);
                         3                3                3
           iI := [x[1]  - x[1], x[2]  - x[2], x[3]  - x[3]]
```

We can now define the functions that associate to the logical connectives their polynomial expressions, as done in *CoCoA* above.

```
NEG :=(m::algebraic) -> NormalForm(2-m,iI,Orde):
NEC :=(m::algebraic) -> NormalForm(expand(m^2+2*m),iI,Orde):
POS :=(m::algebraic) -> NormalForm(expand(2*m^2),iI,Orde):
`&AND`:=(m::algebraic,n::algebraic) ->
        NormalForm(expand(2*m^2*n^2+2*m^2*n+2*m*n^2+m*n),
                   iI,Orde):
`&OR` :=(m::algebraic,n::algebraic) ->
        NormalForm(expand(m^2*n^2+m^2*n+m*n^2+2*m*n+m+n),
                   iI,Orde):
`&IMP` :=(m::algebraic,n::algebraic) ->
        NormalForm(expand(2*m^2*n^2+2*m^2*n+2*m*n^2+m*n+2*m+2),
                   iI,Orde):
`&IFF` :=(m::algebraic,n::algebraic) ->
        NormalForm(expand(m^2*n^2+m^2*n+m*n^2+2*m*n+2*m+2*n+2),
                   iI,Orde):
```

## 12 Description of the shell

A GUI was presented at FLINS'2008 conference [24], and a complete shell is now provided. When working with the shell we could distinguish three levels.

At the lower level, we provide the CAS code for performing knowledge extraction and verification in different logics (i.e., we provide the algebraic inference engines). For instance, to work with the CAS *CoCoA 4.3* in Boolean logic with the propositional variables $x_1, \dots, x_5$, the code that contains the algebraic model of the IE should be included in the file named `CODE_CoCoA.TXT`. It would look like:

```
USE Z/(2)[x[1..5]];
MEMORY.I:=Ideal([x[K_]^2-x[K_] | K_ In 1..5]);
NEG(M):=NF(1+M,MEMORY.I);
OR(M,N):=NF(M+N+M*N,MEMORY.I);
...
```

(similar to the code shown in Section 11.2). Some lines regarding I/O between the GUI and the CAS follow.

At the intermediate level, the RBES developer details the concrete potential facts, rules and integrity constraints of a certain RBES (the code must be stored in file `RBES.TXT`).

**Example 10** *For instance, the production rules and IC of a tiny RBES whose underlying logic is classic Boolean and their CoCoA translations can be (note that in a real RBES there would be dozens of rules):*

| | |
|---|---|
| $R1: \ x[1] \to x[2]$ | `R1:=IMP(x[1],x[2]);` |
| $R2: \ x[2] \wedge \neg x[3] \to x[4]$ | `R2:=IMP(AND(x[2],NEG(x[3])),x[4]);` |
| $R3: \ x[1] \wedge x[4] \to x[5]$ | `R3:=IMP(AND(x[1],x[4]),x[5]);` |
| $NIC1: \neg(x[3] \wedge x[5])$ | `NIC1:=NEG(AND(x[3],x[5]));` |

At the upper level, the final user does not have to deal directly with the CAS but with a simple GUI. Both the number of truth values of the logic (e.g., 3) and the desired logic (e.g., Kleene's) are to be chosen beforehand. Then consistency can be checked and knowledge extraction performed. To do so, a list formed by a proposition and a (consistent) list of facts has to be introduced. The GUI calls the CAS and checks firstly if they lead to inconsistency. Afterwards, it checks if the given proposition follows from the given set of facts.

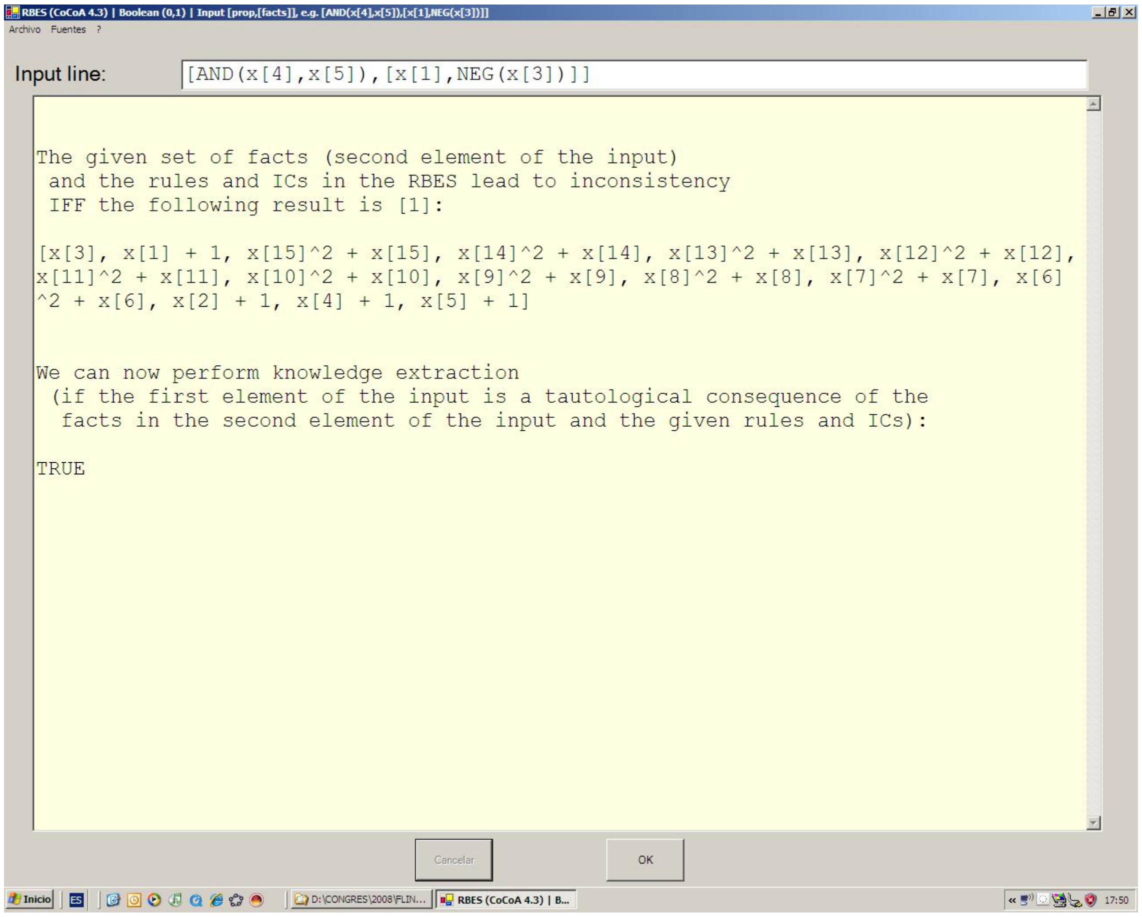
**Example 11** *Let us consider the tiny RBES of Example 10. If the list of given facts is $[x[1], \neg x[3]]$, an inconsistency is not obtained. Moreover, $x[4] \wedge x[5]$ follows from them (see Figure 4).*

This GUI has been implemented in *Visual-BASIC* and runs on computers using the most common windows-based operating system. Porting it to other operating systems is under consideration now. It can call (at least) the mathematical systems: *CoCoA*, *Maple*, *Maxima*, *Singular*, *Risa-Asir* and *Octave*.

## 13 Conclusions

We believe that this algebraic approach is flexible and powerful (as several logics can be easily and effectively simulated) and that the whole shell now put together can be really useful for teaching and for quick RBES designing. This approach has the advantage over that of Kapur-Narendran [15] and Hsiang [13] for Boolean logic (or Alonso et al. [3] and Chazarain et al. [7] in the multi-valued case) of providing an algebraic structure that is isomorphic to the propositional algebra, instead of only performing effective computations in logic using Gröbner bases. This is the key for going one step further and obtaining an algebraic structure isomorphic to a RBES based on one of these logics, i.e., from

$$\mathcal{A} = \mathbb{Z}_p[x, y, \dots, z]/\langle x^p - x, y^p - y, \dots, z^p - z\rangle$$


RBES (CoCoA 4.3) | Boolean (0,1) | Input [prop,[facts]], e.g. [AND(x[4],x[5]),[x[1],NEG(x[3])]]
Archivo Fuentes ?
Input line:
Cancelar
OK

```
[AND(x[4],x[5]),[x[1],NEG(x[3])]]
```

```
The given set of facts (second element of the input)
 and the rules and ICs in the RBES lead to inconsistency
 IFF the following result is [1]:

[x[3], x[1] + 1, x[15]^2 + x[15], x[14]^2 + x[14], x[13]^2 + x[13], x[12]^2 + x[12],
x[11]^2 + x[11], x[10]^2 + x[10], x[9]^2 + x[9], x[8]^2 + x[8], x[7]^2 + x[7], x[6]
^2 + x[6], x[2] + 1, x[4] + 1, x[5] + 1]


We can now perform knowledge extraction
 (if the first element of the input is a tautological consequence of the
  facts in the second element of the input and the given rules and ICs):

TRUE
```

Figure 4. Screenshot of a GUI for RBES consistency checking and knowledge extraction.

to

$$\mathcal{A}/J$$

where $J$ is the ideal generated by the negation of the facts, rules and integrity constraint, so that knowledge extraction and verification of RBES can be performed.

**Acknowledgement.** This work was partially supported by the research projects *TIN2006-06190* (Government of Spain) and *UCM2008-910563* (*UCM - BSCH Gr. 58/08*, Research Group *ACEIA*, Spain).

We would also like to thank the anonymous referees for their most valuable comments and suggestions.

## References

[1] ADAMS, W. W. AND LOUSTAUNAU, P., (1994). *An Introduction to Gröbner Bases*. Graduate Studies in Mathematics **3**, AMS.

[2] AKRITAS, A. G., (1989). *Elements of Computer Algebra with Applications*, Wiley - Interscience.

[3] ALONSO, J. A. AND BRIALES, E., (1989). Lógicas Polivalentes y Bases de Gröbner, en M. Vide ed., *Procs. of the V Congress on Natural Languages and Formal Languages*, Barcelona Univ. Press, 307–315.

[4] BUCHBERGER, B., (1988). Applications of Gröbner Bases in non-linear Computational Geometry. In J. R. Rice (editor), *Mathematical Aspects of Scientific Software*, IMA Volumes in Math. and its Applications, **14**, Springer-Verlag.

[5] BUCHBERGER, B., (2006). Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elementals of the residue class ring of a zero dimensional polynomial ideal, *J. Symbolic Comput.*, **41**, 3-4, 475–511. DOI: 10.1016/j.jsc.2005.09.007

[6] CAPANI, A. AND NIESI, G., (1996). *CoCoA User's Manual (v. 3.0b)*, Depto. de Matemáticas, Universidad de Genova.

[7] CHAZARAIN, J.; RISCOS, A.; ALONSO, J. A. AND BRIALES, E., (1991). Many-valued Logic and Gröbner Bases with Applications to Modal Logic, *J. Symbolic Comput.*, **11**, 181–194.

[8] CHOU, S.-C., (1987). *Mechanical Geometry Theorem Proving*. Kluwer Academic Publishers.

[9] COX, D.; LITTLE, J. AND O'SHEA, D., (1992). *Ideals,varieties, and algorithms*, Springer-Verlag.

[10] HALMOS, P. R., (1974). *Lectures on Boolean Algebras*, Springer-Verlag.

[11] HALMOS, P. AND GIVANT, S., (1998). *Logic as Algebra*, The Mathematical Asociation of America.

[12] HERMES, H., (1963). *La teoría de retículos y su aplicación a la lógica matemática*, Conf. Mat. VI., CSIC-Madrid.

[13] HSIANG, J., (1985). Refutational Theorem Proving using Term-rewriting Systems, *Artificial Intelligence*, **25**, 255–300. DOI: 10.1016/0004-3702(85)90074-8

[14] KAPUR, D. AND MUNDY, J. L., (1989). Wu's method and its application to perspective viewing. In D. Kapur and J. L. Mundy, eds., *Geometric Reasoning*. MIT Press, Cambridge MA, 15–36.

[15] KAPUR, D. AND NARENDRAN, P., (1984). An Equational Approach to Theorem Proving in First-Order Predicate Calculus, 84CRD296, *General Electric Corporate Research and Development Report* (Schenectady, NY, March 1984, rev Dec 1984). Also in A. K. Joshi, ed., *Proceedings of IJCAI-85*. Morgan Kaufmann, 1146–1153.

[16] KREUZER, M. AND ROBBIANO, L., (2000). *Computational Commutative Algebra*. Springer-Verlag.

[17] LAITA, L. M. AND DE LEDESMA, L., (1997). Knowledge-Based Systems Verification. In Kent, J. G. Williams (editors), *Encyclopedia of Computer Science and Technology*. Marcel Dekker, 253–280.

[18] LAITA, L. M.; DE LEDESMA, L.; ROANES-LOZANO, E. AND ROANES-MACÍAS, E., (1995). An Interpretation of the Propositional Boolean Algebra as a $k$-algebra. Effective Calculus. In J. Campbell, J. Calmet (editors), *Proceedings of AISMC-2*, Springer-Verlag LNCS 958, 255–263.

[19] LAITA, L. M.; ROANES-LOZANO, E.; DE LEDESMA, L. AND ALONSO, J. A., (1999). A Computer Approach to Verification and Deduction in Many-valued Knowledge Systems, *Soft Comput.*, **3**, 1, 7–19. DOI: 10.1007/s005000050086

[20] LAITA, L. M.;ROANES-LOZANO, E.; MAOJO, V.; DE LEDESMA, L. AND LAITA, L., (2001). An expert system for managing medical appropriateness criteria based on computer algebra techniques, *Comput. Math. Appl.*, **42**, 5, 12, 1505–1522. DOI: 10.1016/S0898-1221(01)00258-9

[21] MENDELSON, E., (1970). *Theory and Problems of Boolean Algebras and Switching Circuits*, Schaum's, MacGraw-Hill.

[22] MONK, D., (1989). *Handbook of Boolean Algebras*, North-Holland.

[23] RECIO, T. AND VÉLEZ, M. P., (1999). Automatic Discovery of Theorems in Elementary Geometry, *J. Automat. Reason.*, **23**, 63–82.

[24] ROANES-LOZANO, E.; HERNANDO, A.; LAITA, L. M. AND ROANES-MACÍAS, E., (2008). A Shell for Rule-Based Expert Systems Development Using Groebner Bases-Based Inference. In D. Ruan, J. Montero, J. Lu, L. Martínez, P. D'Hondt, E. E. Kerre (editors), *Computational Intelligence in Decision and Control. Proceedings of the 8th International FLINS Conference*, World Scientific Proceedings Series on Computer Engineering and Information Science **1**, 769–774. DOI: 10.1142/9789812799470_0126

[25] Roanes-Lozano, E.; Hernando, A.; Laita, L. M. and Roanes-Macías, E., A Groebner Bases-based Approach to Backward Reasoning in Rule Based Expert Systems, *Ann. Math. Artif. Intell.*, to appear. DOI: 10.1007/s10472-009-9147-4

[26] Roanes-Lozano, E.; Laita, L. M. and Roanes-Macías, E., (1995). Maple V in A.I., The Boolean Algebra Associated to a KBS, *CAN Nieuwsbrief*, **14**, 65–70.

[27] Roanes-Lozano, E.; Laita, L. M. and Roanes-Macías, E., (1998). A Polynomial Model for Many-valued Logics with a Touch of Algebraic Geometry and Computer Algebra, *Math. Comput. Simulation*, **45**, 1 83–99.

[28] Roanes-Lozano, E.; Laita, L. M. and Roanes-Macías, E., (2003). Cálculos efectivos en Lógica Proposicional Booleana interpretada como un anillo de clases residuales (polinomial) sobre $\mathbb{Z}_2$, *Boletín de la Sociedad "Puig Adam" de Profesores de Matemáticas*, **65**, 17–42.

[29] Roanes-Lozano, E.; Laita, L. M. and Roanes-Macías, E., (2008). A Groebner Bases Based Many-valued Modal Logic Implementation in Maple. In S. Autexier et al. (editors), *AISC / Calculemus / MKM 2008*, LNAI 5144 Springer-Verlag, 170–183. DOI: 10.1007/978-3-540-85110-3_14

[30] Roanes-Lozano, E.; Muga, R.; Laita, L. M. and Roanes-Macías, E., (2005). A terminal area topology-independent GB-based conflict detection system for A-SMGCS, *RACSAM Rev. R Acad. Cienc. Exactas, Fís. Nat. Ser. A Mat.*, **98**, 1-2, 229–237.

[31] Roanes-Lozano, E.; Roanes-Macías, E. and Laita, L. M., (1999). Geometric Interpretation of Strong Inconsistency in Knowledge Based Systems. In V. G. Ganzha, E. W. Mayr, E. V. Vorozhtsov (editors), *Computer Algebra in Scientific Computing. Proceedings of CASC'99).* Springer-Verlag, 349–363.

[32] Roanes-Lozano, E.; Roanes-Macías, E. and Laita, L. M., (2000). Railway Interlocking Systems and Groebner Bases, *Math. Comput. Simulation*, **51**, 5, 473–481. DOI: 10.1016/S0378-4754(99)00137-8

[33] Roanes-Lozano, E.; Roanes-Macías, E. and Laita, L. M., (2004). The Geometry of Algebraic Systems and Their Exact Solving Using Groebner Bases, *Computing in Science and Engineering*, **6**, 2, 76–79. DOI: 10.1109/MCISE.2004.1267612

[34] Roanes-Lozano, E.; Roanes-Macías, E. and Laita, L. M., (2004) Some Applications of Gröbner Bases, *Computing in Science and Engineering*, **6**, 3, 56–60. DOI: 10.1109/MCISE.2004.1289309

[35] Robbiano, L. et al., (2002). *CoCoA 4.2 Online Help*, (electronic file companying CoCoA v.4.2).

[36] Stone, M., (1940). The Theory of Representations for Boolean Algebras, *Trans. Amer. Math. Soc.*, **40**, 1, 37–111. DOI: 10.2307/1989664

[37] Turner, R., (1984). *Logics for Artificial Intelligence*, Ellis Horwood.

[38] Winkler, F., (1996). *Polynomial Algorithms in Computer Algebra*. Springer-Verlag.

**Eugenio Roanes-Lozano**
Depto. de Álgebra
Facultad de Educación
Universidad Complutense de Madrid
c/ Rector Royo Villanova s/n
28040-Madrid
Spain
`eroanes@mat.ucm.es`

**Luis M. Laita**
Depto. CC. de la Computación e I.A.
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo
Boadilla del Monte, 28660-Madrid
Spain
`laita@fi.upm.es`

**Antonio Hernando**
Depto. de Sistemas Inteligentes Aplicados
Escuela Universitaria de Informática
Universidad Politécnica de Madrid
Carretera de Valencia km 7
28031-Madrid
Spain
`ahernando@eui.upm.es`

**Eugenio Roanes-Macías**
Depto. de Álgebra
Facultad de Educación
Universidad Complutense de Madrid
c/ Rector Royo Villanova s/n
28040-Madrid
Spain
`roanes@mat.ucm.es`