FROM COMPUTER ALGEBRA TO DISCRETIZED CONTINUOUS LOGIC

Eugenio Roanes-Lozano[1] **Javier Montero**[2] **Antonio Hernando**[3] **Luis M. Laita**[4]

[1] Depto. de Álgebra, Universidad Complutense de Madrid, eroanes@mat.ucm.es
[2] Depto. de Estadística e Investigación Operativa, Universidad Complutense de Madrid, monty@mat.ucm.es
[3] Depto. de Sistemas Inteligentes Aplicados, Universidad Politécnica de Madrid, ahernando@eui.upm.es
[4] Depto. de Inteligencia Artificial, Universidad Politécnica de Madrid, laita@fi.upm.es

Abstract

The aim of this paper is to present a new algebraic approach from computer algebra to a discretized continuous logic. It makes use of a previous model of p-valued logic (where p is a prime number) based on the use of Gröbner bases of polynomial ideals. A five-valued logic (i.e., $p = 5$) with some modal operators has been considered as a compromise between precision and complexity of the polynomials involved. Therefore the continuous truth values are discretized into five intervals corresponding to the likelihood levels: impossible or very unlikely / unlikely / dubious / probably / almost sure or absolutely sure. It is therefore posible to obtain the likelihood level of any given logic formula. Moreover, it is possible to perform knowledge extraction and verification of small Rule Based Expert System whose knowledge is represented by this logic. An implementation in the computer algebra system *Maple* is included.

Palabras Clave: Continuous Logic, Computer Algebra, Gröbner Bases, Rule Based Expert Systems.

1 INTRODUCTION

In previous research, the authors have developed a polynomial model for knowledge extraction and consistency checking in Rule Based Expert Systems (RBES). The underlying logic of such a polynomial model is the classical Boolean logic or finitely-valued logics with modal operators [12, 13]. In particular, the proposed model is based on the theory of Gröbner bases (GB) [2, 3]. Following this approach, the authors have so far designed and developed RBES in different fields [9, 11, 15]. These works are summarized in Section 2.

In [14] the authors developed an algebraic approach to minimal polynomial continuous logic (MPL) that can be applied to perform knowledge extraction and verification (consistency checking) of general RBES whose underlying logic is MPL, but, unfortunately, it only provides partial results and sufficient conditions, unlike the Boolean and finitely-valued cases.

An implementation of any continuous logic, fuzzy logic in particular, presents serious computational and estimation difficulties. The aim of this paper is to perform a logical effective calculations and to perform knowledge extraction and consistency checking in RBES whose underlying logic is continuous, by means of a discretized continuous logic.

In this way, our research extends previous works by Kapur and Narendran [8] and Hsiang [7] (classical Boolean logic case) and Alonso et al. [1, 4] finitely-valued logics case), where techniques for performing effective calculations in logic using Gröbner bases are treated.

2 PREVIOUS RESULTS

2.1 POLYNOMIAL IDEALS AND GRÖBNER BASES

A polynomial ideal is a subset of a polynomial ring which fulfils some specific requirements: it is also a ring and the product of any element of the ideal by any element of the ring lies within the ideal. The ideal generated by the polynomials, $p_1,...,p_m$ (this ideal is mathematically denoted by $\langle p_1,...,p_m\rangle$) is the minimum ideal containing $p_1,...,p_m$ [5].

The key result in ideal theory is due to Buchberger [2, 3, 5]. It happens that different bases may generate the same idealn and Buchberger designed a theory and an algorithm for finding a specific basis, which he called "Gröbner basis" (as a tribute to his PhD advisor), that characterized each polynomial ideal [2, 3, 5]. A constructive method for calculating the "normal form" (NF) of a polynomial modulo an ideal (the residue of the polynomial modulo the ideal) was also included in the theory. The most relevant application of Buchberger approach is the solution of the "ideal membership problem": if g is a polynomial and L is an ideal, then

$$g \in L \text{ if and only if } NF(g, L) = 0$$

2.2 A POLYNOMIAL MODEL FOR FINITELY-VALUED MODAL LOGIC

Let us suppose that the logic considered is a p-valued one (with some modal operators) and that the propositional variables are P_1,..., P_w. Then the polynomial residue class ring

$$\mathbb{Z}_p[p_1, ..., p_w]/I$$

where I is the ideal

$$I = \langle p_i^p, ..., p_w^p \rangle$$

with certain operations translating the connectives of the logic is isomorphic to the p-valued logic. Given a propositional formula, α, we use $\varphi(\alpha)$ to denote the polynomial representing α (in the Boolean case this representation is closely related to Boole's mathematical approach to logic). Details about this translation in the p-valued case can be found in [13]. According to Theorem 1 (below), the problem of checking if a propositional formula α can be inferred (formally termed as "tautological consequence") from others, β_1,..., β_m, may be dealt by checking a polynomial ideal membership:

Theorem 1: A propositional formula α is a tautological consequence of a set of formulae $\{\beta_1,..., \beta_m\}$, if and only if

$$\varphi(\neg\alpha) \in \langle \varphi(\neg\beta_1), ..., \varphi(\neg\beta_m) \rangle$$

A long detailed proof of this beautiful theorem (in the p-valued case) can be found in [13].

2.3 RULE-BASED EXPERT SYSTEMS

A Rule-Based Expert System (RBES) consists of three basic components:

- The *Knowledge Base*. It is concerned with the information contained in the Expert System trying to model the knowledge of human experts. In a RBES, this knowledge is characterized by means of a set of production rules which are used along with the input of the RBES to derive the output of the system. The process of developing this knowledge base requires the choice of a representation paradigm for modeling all the information described in natural language by human experts. In the case of the method described in the present paper, the suitable information is represented by means of polynomials. Consequently, in our system, the information related to the input (facts), output and the knowledge base (production rules) must be translated to polynomials. In order to make easy this translation, we are previously required to represent first all this information in terms of propositional logic.
- The *inference engine*. It is related to the technique used to make deductions automatically (that is to say, the mechanism by means of which the Expert System derives the output from the given input). By means of a previous mathematical result (see Theorem 1), problems associated to "deduction" may be translated into algebraic problems, so that we can use a mathematical algebraic software as inference engine.
- An interactive *Graphic User's Interface*. Through it, users may easily introduce the information concerning the input of the Expert System and observe the resulting output the Expert System automatically infers from the former.

2.4 A POLYNOMIAL MODEL FOR RULE-BASED EXPERT SYSTEMS

In the case of a RBES, Theorem 1 can be formulated in the following way:

Theorem 2: A certain formula q can be inferred from the knowledge in the RBES (described by the production rules R_1,..., R_r) and the facts $\{F_1,..., F_f\}$ if and only if:

$$\varphi(\neg q) \in \langle \varphi(\neg R_1), ..., \varphi(\neg R_r), \varphi(\neg F_1), ..., \varphi(\neg F_f) \rangle$$

In order to simplify this expression, we can define the ideal J, generated by the production rules in the knowledge base,

$$J = \langle \varphi(\neg R_1), ..., \varphi(\neg R_r) \rangle$$

and the ideal K generated by the given facts,

$$K = \langle \varphi(\neg F_1), ..., \varphi(\neg F_f) \rangle$$

By means of these two ideals, the expression above would be transformed into:

$$\varphi(\neg q) \in J + K$$

and the latter membership can, again, be decided using normal forms, as it is equivalent to:

$$NF(\neg(q), J + K) = 0$$

if working in $\mathbb{Z}_p[p_1, ..., p_w]/I$, or to:

$$NF(\neg(q), I + J + K) = 0$$

if working in $\mathbb{Z}_p[p_1, ..., p_w]$. Knowledge extraction in this kind of RBES can be therefore effectively calculated. Moreover, $(\mathbb{Z}_p[p_1, ..., p_w]/I)/(J + K)$ is in fact isomorphic to the logic structure associated to the RBES (see [12] for details). Consequently, consistency checking for a given set of facts $\{\varphi(\neg F_1), ..., \varphi(\neg F_f)\}$ is equivalent to the non-degeneracy of ideal $J + K$ of $(\mathbb{Z}_p[p_1, ..., p_w]/I)$ into the whole ring, what is equivalent to:

$$GB(J + K) \neq \{1\}$$

if working in $\mathbb{Z}_p[p_1, ..., p_w]/I$, or to:

$$GB(I + J + K) \neq \{1\}$$

if working in $\mathbb{Z}_p[p_1, ..., p_w]$.

2.5 MAPLE IMPLEMENTATION

The implementation in a computer algebra system *Maple*, *CoCoA*, etc. is surprisingly brief. For instance, all the code corresponding to Kleene's five-valued logic (with some modal operators) is included afterwards. Firstly the polynomial variables, ring, variable ordering and ideal iI have to be defined (I is a reserved word in *Maple*, so we denote ideal I by iI):

```
> with(Groebner):
> with(Ore_algebra):
> SV:=x[1],x[2],x[3],x[4]:
> fu:=v->v^5-v:
> A:=poly_algebra(SV,characteristic=5):
> Orde:=MonomialOrder(A,'plex'(SV)):
> iI:=map(fu,[SV]):
```

and then the unary and binary connectives can be defined:

```
> NEG  :=(m::algebraic)  ->
>        NormalForm(4*m+4,iI,Orde):
> POS1 :=(m::algebraic)  ->
>        NormalForm(expand(4*m^4),iI,Orde):
> POS2 :=(m::algebraic)  ->
>        NormalForm(expand(3*m^4+4*m^3+4*m^2
>        +4*m),iI,Orde):
> NEC2 :=(m::algebraic)  ->
>        NormalForm(expand(2*m^4+2*m^3+m),
>        iI,Orde):
> NEC1 :=(m::algebraic)  ->
>        NormalForm(expand(m^4+4*m^3+m^2+
>        4*m),iI,Orde):
> `&OR`:=(m::algebraic,n::algebraic) ->
>        NormalForm(expand(2*m^4*n^2+
>        4*m^3*n^3+2*m^2*n^4+2*m^4*n+2*m*n^4+
>        m^3*n+3*m^2*n^2+m*n^3+2*m*n+m+n),
>        iI,Orde):
> `&AND` :=(m::algebraic,n::algebraic) ->
>        NormalForm(expand(3*m^4*n^2+m^3*n^3+
>        3*m^2*n^4+3*m^4*n+3*m*n^4+4*m^3*n+
>        2*m^2*n^2+4*m*n^3+3*m*n),iI,Orde):
> `&IMP` :=(m::algebraic,n::algebraic) ->
>        NEG(m) &OR n:
```

3 DISCRETIZED CONTINUOUS LOGIC

In this section we present an algebraic approach to a discretized continuous logic.

Our approach is straightforward once we have a model for finitely-valued logics, as shown below. Notice that at a first stage we shall in fact define a crisp partition of the unit interval, leaving for a future research the case with fuzzy classes, either in the sense of Ruspini [16] or Montero *et al.* [6, 10]. At this stage, five likelihood levels will be considered in the discretized continuous logic:

- impossible or very unlikely,
- unlikely,
- dubious,
- probably,
- almost sure or absolutely sure.

The continuous truth values are discretized as follows:

- $[0, 1/5) \rightsquigarrow$ impossible or very unlikely $\rightsquigarrow$ $\rightsquigarrow$ 0 in the five-valued logic,
- $[1/5, 2/5) \rightsquigarrow$ unlikely $\rightsquigarrow$ $\rightsquigarrow$ 1 in the five-valued logic,
- $[2/5, 3/5) \rightsquigarrow$ dubious $\rightsquigarrow$ $\rightsquigarrow$ 2 in the five-valued logic,
- $[3/5, 4/5) \rightsquigarrow$ probably $\rightsquigarrow$ $\rightsquigarrow$ 3 in the five-valued logic,
- $[4/5, 1] \rightsquigarrow$ almost sure or absolutely sure $\rightsquigarrow$ $\rightsquigarrow$ 4 in the five-valued logic.

Given a certain logic formula and the numerical truth values of the propositional variables in the logic formula (ranging in $[0,1]$), the process to obtain the likelihood level of the formula in the discretized continuous logic is the following:

1) compute the polynomial corresponding to the logic formula,

2) calculate the the corresponding truth values of the propositional variables in the five-valued logic (ranging in $\{0,1,2,3,4\}$), as detailed above,

3) substitute the latter truth values in the polynomial,

4) find to which likelihood level the truth value obtained for the formula corresponds.

Let us note that, once the values are discretized, the value obtained for disjunction and conjunction correspond to maximum and minimum (if one of the usual finitely-valued logics such as Łukasiewicz's or Kleene's are chosen). For instance, the truth table of the connective "conjunction"is, in the five-valued case:

$\wedge$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	2	2	2
3	0	1	2	3	3
4	0	1	2	3	4

3.1 IMPLEMENTATION

The code included in Section 2.5 can be reused for this discretization of a continuous logic. Two more procedures have to be added:

- `tVal` substitutes the numerical values in the polynomial translating the logic formula (steps 2) and 3) of Section 3),
- `evalFu` finds to which likelihood level the truth value obtained for the formula corresponds (step 4) of Section 3).

Truth value introduction is done in *Maple* as follows: if variable x[2] has a truth value 0.65 we assign:

```
> val(x[2]):=.65;
```

Let us include a simple example afterwards:

```
> val(x[1]):=.3:
> val(x[2]):=.1:
> val(x[3]):=.75:
> forml:=x[1] &OR (x[2] &OR x[3]):
> evalFu(tVal(forml));
                Probably
```

4 AN ALGEBRAIC APPROACH TO RBES

The algebraic model is the same as in the previous section, but, instead of computing likelihood levels from truth values, we perform knowledge extraction (and verification). In particular, in the above five-valued logic we have that

- $\Box_1 f$ is true iff f has the truth value 4 (true),
- $\Box_2 f$ is true iff f has a truth value ≥ 3,
- $\Diamond_2 f$ is true iff f has a truth value ≥ 2,
- $\Diamond_1 f$ is true iff f has a truth value ≥ 1,
- $\Diamond_1 f \vee \Diamond_1 \neg f$ is always true.

as can be checked in the following truth table

f	$\Diamond_1 f \vee \Diamond_1 \neg f$	$\Diamond_1 f$	$\Diamond_2 f$	$\Box_2 f$	$\Box_1 f$
0	4	0	0	0	0
1	4	4	0	0	0
2	4	4	4	0	0
3	4	4	4	4	0
4	4	4	4	4	4

Consequently,

- $\Box_1 f$ is true iff f has the truth value 4,
- $\Box_2 f \wedge \Diamond_1 \neg f$ is true iff f has the truth value 3,
- $\Diamond_2 f \wedge \Diamond_2 \neg f$ is true iff f has the truth value 2,
- $\Box_2 \neg f \wedge \Diamond_1 f$ is true iff f has a truth value 1,
- $\Box_1 \neg f$ is true iff f has the truth value 0.

as can be checked in the following truth table

f	$\Box_1 f$	$\Box_2 f \wedge$ $\Diamond_1 \neg f$	$\Diamond_2 f \wedge$ $\Diamond_2 \neg f$	$\Box_2 \neg f$ $\wedge \Diamond_1 f$	$\Box_1 \neg f$
0	4	0	0	0	0
1	0	4	0	0	0
2	0	0	4	0	0
3	0	0	0	4	0
4	0	0	0	0	4

Therefore, when translating the rules from the experts' vocabulary into logic, it has to be considered for which truth values we want the rule to be fired.

Example 1: For instance, the rule $R1$:

$$\Box_2(q) \to g$$

is fired if

the likelihood level of q is "probably" or "almost sure or absolutely sure"

i.e., if

q has the truth values 3 or 4 in the five-valued logic.

Once discretized, knowledge extraction and consistency checking can be performed using normal forms and Gröbner bases, exactly as in Section 2.4.

4.1 EXAMPLES IN MAPLE

Example 2: Let us check Example 1 with *Maple*: 0 is obtained as the normal form of g with respect to the ideal generated by a base of ideal I, the negation of the antecedent and a base of ideal J, when the antecedent is:

$$\Box_1 f \quad \text{or} \quad \Box_2 f \wedge \Diamond_1 \neg f$$

and not when the antecedent is:

$$\Diamond_2 f \wedge \Diamond_2 \neg f \quad \text{or} \quad \Box_2 \neg f \wedge \Diamond_1 f \quad \text{or} \quad \Box_1 \neg f$$

as shown afterwards:

```
> R1:=NEC2(q) &IMP g:
> J:=[NEG(R1)]:
> B:=Basis([op(iI),NEG(NEC1(q)),op(J)],Orde):
> NormalForm(NEG(g),B,Orde);
                                   0
> B:=Basis([op(iI),NEG(NEC2(q) &AND
>          POS1(NEG(q))),op(J)],Orde):
> NormalForm(NEG(g),B,Orde);
                                   0
> B:=Basis([op(iI),NEG(POS2(q) &AND
>          POS2(NEG(q))),op(J)],Orde):
> NormalForm(NEG(g),B,Orde);
                                4 g + 4
> B:=Basis([op(iI),NEG(NEC2(NEG(q)) &AND
>          POS1(q)),op(J)],Orde):
> NormalForm(NEG(g),B,Orde);
                                4 g + 4
> B:=Basis([op(iI),NEG(NEC1(NEG(q))),op(J)],
>          Orde):
> NormalForm(NEG(g),B,Orde);
                                4 g + 4
```

Example 3: Let us consider the following tiny 10-rules RBES (we already include the *Maple* code, in order to show its simplicity):

```
> R1:=NEC1(q) &IMP p2:
> R2:=q &IMP p3:
> R3:=r &IMP p2:
> R4:=r &IMP p3:
> R5:=(q &AND x) &IMP g:
> R6:=NEC1(s) &IMP x:
> R7:=s &IMP p3:
> R8:=u &IMP g:
> R9:=v &IMP p1:
> R10:=w &IMP p2;
```

Therefore, the ideal of rules (J) is:

```
> J:=[ NEG(R1),NEG(R2),NEG(R3),NEG(R4),
>      NEG(R5),NEG(R6),NEG(R7),NEG(R8),
>      NEG(R9),NEG(R10)]:
```

Let us state as true (i.e., consider the facts):

$$\Box_1(s) \quad \text{and} \quad q.$$

Then the ideal of facts (K) is:

```
> K:=[NEG(NEC1(s)),NEG(q)]:
```

Now, do x and g follow from these facts? Let us first check the consistency for this set of facts, i.e., that ideal $I + J + K$ is not the whole ring (i.e., that the corresponding Gröbner basis is not `[1]`):

```
> B:=Basis([op(iI),op(J),op(K)], Orde);
  B := [1 + g, 1 + p3, p2 + 1, ...]
```

Now we can check whether x and g follow form this facts:

```
> NormalForm(NEG(x),B,Orde);
                                   0
> NormalForm(NEG(g),B,Orde);
                                   0
```

so both of them do follow.

Example 4: Let us consider the same tiny RBES and let us now state as true (i.e., consider the facts):

$$\Box_2(s) \wedge \Diamond_1(\neg(s)) \quad \text{and} \quad q.$$

Then the ideal of facts (K) is:

```
> K:=[NEG(NEC2(s) &AND POS1(NEG(s))),NEG(q)];
```

Now, do x and g follow from these facts? The following Gröbner basis is not `[1]`, so there is consistency for this set of facts:

```
> B:=Basis([op(iI),op(J),op(K)], Orde);
        5
  B := [g  + 4 g, 1 + p3, p2 + 1, ...]
```

And:

```
> NormalForm(NEG(x),B,Orde);
                                        4 x + 4
> NormalForm(NEG(g),B,Orde);
                                        4 g + 4
```

are not 0, so neither x nor g follow from this set of facts.

5 CONCLUSIONS

The approach presented in this paper provides a first computational frame for performing effective computations (knowledge extraction and verification) in RBES whose underlying logic is a continuous logic, by means of its discretization. The drawback is the complexity of the polynomials involved, that only allow to treat medium-size RBES (in the Boolean case RBES with figures like a hundred rules can be treated in a standard computer in a matter of seconds). Future research will consider a more sophisticated discretization approach in order to implement alternative modal or fuzzy logics.

Acknowledgements

This work was partially supported by grants TIN2009-07901 (Government of Spain) and UCM2008-910563 (UCM Research Group 58/08, ACEIA).

References

[1] J. A. Alonso, E. Briales: Lógicas Polivalentes y Bases de Gröbner. In: C. Martin, ed.: *Actas del V Congreso de Lenguajes Naturales y Lenguajes Formales*. University of Seville, Seville (1995) 307–315.

[2] B. Buchberger: Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elementals of the residue class ring of a zero dimensional polynomial ideal. *Journal of Symbolic Computation* 41/3-4 (2006) 475-511.

[3] B. Buchberger: Applications of Gröbner Bases in Non-Linear Computational Geometry. In: J. R. Rice, ed.: *Mathematical Aspects of Scientific Software*. Springer-Verlag, IMA Vol. 14, New York (1988) 60–88.

[4] J. Chazarain, A. Riscos, J. A. Alonso, E. Briales: Multivalued Logic and Gröbner Bases with Applications to Modal Logic. *Journal of Symbolic Computation* 11 (1991) 181–194.

[5] D. Cox, J. Little, D. O'Shea: Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, New York, 1992.

[6] A. Del Amo, J. Montero, G. Biging, V. Cutello: Fuzzy classification systems. *European Journal of Operational Research* 156 (2004) 459-507.

[7] J. Hsiang: Refutational Theorem Proving using Term-Rewriting Systems. *Artificial Intelligence* 25 (1985) 255–300.

[8] D. Kapur, P. Narendran: An Equational Approach to Theorem Proving in First-Order Predicate Calculus. In: *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*, vol. 2 (1985) 1146–1153.

[9] L.M. Laita, E. Roanes-Lozano, V. Maojo, L. de Ledesma, L. Laita: An Expert System for Managing Medical Appropriateness Criteria Based on Computer Algebra Techniques. *Computers and Mathematics with Applications* 51/5 (2000) 473–481.

[10] J. Montero, D. Gómez, H. Bustince: On the relevance of some families of fuzzy sets. *Fuzzy Sets and Systems* 158 (2007) 2429-2442.

[11] C. Pérez-Carretero, L.M. Laita, E. Roanes-Lozano, L. Lázaro, J. González-Cajal, L. Laita: A Logic and Computer Algebra-Based Expert System for Diagnosis of Anorexia. *Mathematics and Computers in Simulation* 58 (2002) 183–202.

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

[13] E. Roanes-Lozano, L. M. Laita and E. Roanes-Macías, A Polynomial Model for Multivalued Logics with a Touch of Algebraic Geometry and Computer Algebra. *Mathematics and Computers in Simulation* 45/1 (1998) 83–99.

[14] E. Roanes-Lozano, L.M. Laita, E. Roanes-Macías, A Symbolic-Numeric Approach to MPL Continous Logic and to Rule Based Expert Systems whose Underlying Logic is MPL. *The Open Applied Mathematics Journal* 1 (2008) 126-133.

[15] C. Rodríguez-Solano, L.M. Laita, E. Roanes-Lozano, L. López-Corral, L. Laita, A Computational System for Diagnosis of Depressive Situations. *Expert System with Applications* 31 (2006) 47–55.

[16] E.H. Ruspini: A new approach to clustering. *Information and Control* 15 (1969) 22-32.