A Computer Algebra Based Knowledge System for Diagnosis and Treatment of Migraine

Luis M. Laita[1], Beatriz López-Bravo[1], Eugenio Roanes-Lozano[2], Jose A. Alonso[3], Luis de Ledesma[1], Víctor Maojo[1], Laura Laita[4]

[1] Universidad Politécnica de Madrid, Depto. de Inteligencia Artificial, Campus de Montegancedo, Boadilla del Monte, E-28660-Madrid, Spain
[2] Universidad Complutense de Madrid, Depto. de Algebra, Edificio "La Almudena", c/ Rector Royo Villanova s/n, E-28040-Madrid, Spain
[3] Universidad de Sevilla, Depto. de CC. de la Computación e Inteligencia Artificial, E.T.S.I. Informática, Avda. Reina Mercedes s/n, E-41012 Sevilla, Spain
[4] Universidad Complutense de Madrid, Escuela de Enfermería, Ciudad Universitaria, E-28040 Madrid, Spain.

Abstract. A Rule-Based Knowledge System (RBKS), that deals with diagnosis and treatment of migraine, and whose inference engine is based in Computer Algebra, is described. The system is built up from three subsystems that deal with the possibility that a patient suffers migraine, the type of migraine the patient may suffer and with the treatment.
Each subsystem has a "knowledge base", containing the necessary information, and an "inference engine", which enables to verify both consistency and extract consequences from the information contained in the knowledge base. Although the RBKS described in this article is complete and even includes a user interface, it is a prototype: the knowledge of migraine contained in its knowledge base is a simplification of experts' knowledge of this complex illness.
The Gröbner bases approach to RBKS described in this article could be applied to more sophisticated knowledge bases on this illness and other illnesses and is even able to deal with multi-valued logics.

Keywords: Medical Diagnosis, Migraine, Rule-Based Knowledge Systems, Computer Algebra

1 Introduction

The design of any Rule-Based Knowledge System, hereinafter denoted as RBKS, requires building a "knowledge base" (to be denoted KB) and an "inference engine". If possible, a "user interface" should also be provided.

The KB contains the experts' knowledge in form of condensed expressions (logical expressions in our case). Apart from literature on the topic [1–4], three human experts were consulted.

The inference engine is a procedure to verify both consistency of the KB and to extract automatically consequences from the information contained in the KB.

We believe that Knowledge or Expert Systems for diagnoses and treatments in medicine cannot, so far, substitute a specialist or a team of specialists. Accordingly, our system has been designed to be not more than a helpful tool for family doctors and sanitary personnel not specialized in migraine, who may match their own opinion about a patient against the "opinion" (the output) of the system, before deciding whether sending the patient to a specialist or not.

Nevertheless the interest of a system like ours is justified because of the following two reasons.

1. A systematic methodology for building a KB for migraine is presented. This methodology may be used, after appropriate but not essential changes, in the study of other illnesses.
2. The inference engine that we have developed is exact and can be used in the study of other illnesses. Its efficiency will be discussed in Section 5.

We have made a few pilot experiences with the system asking two family doctors to use the system to find a tentative diagnosis and a tentative treatment for some persons that, allegedly, suffered migraine. The experience has been positive, in the sense that the system and the users' opinions coincided in most cases. Nevertheless, the two family doctors considered that our KB needed further refinement, in the sense that it should be improved by adding more detailed information. They also insisted on the idea that the final judgement must be always done by a specialist.

Our system is built up from three subsystems. The first subsystem outputs the possibility that a patient suffers migraine (based on his/her predisposing and triggering factors). If this possibility is low or non-existent, the process stops. If the possibility is medium or high, the second subsystem is activated. This second subsystem produces a diagnosis about the type of migraine the patient may suffer. The third subsystem deals with the treatment of migraine. Each one of the three subsystems has a KB and an "inference engine". We have also developed a user interface for the whole RBKS.

Our approach to RBKS, which is founded on "Normal Forms" and "Gröbner bases" [5–7], could be applied to more sophisticated KBs on this illness and other illnesses and is even able to deal with multi-valued Logics.

2 Background

As this article is intended to be read by different audiences, its logical and mathematical background will be described informally. A few explanatory notes will be added when necessary and an outline of the proof of the main theorem is added as Appendix II.

The logic-mathematical ideas to be presented in this section is based on the work of Kapur-Narendran, Hsiang and Chazarain et al. [8–10] and our prior work on the application of Gröbner bases to automated deduction [11, 12], which we applied to the study of medical appropriateness criteria [13], to the diagnosis of anorexia [14] and to other fields (railway interlocking systems) [15].

The method of construction of the anorexia KB differed from the method used in the present migraine study because when we built the former we had not thought of applying Karnaugh diagrams [16], which simplify the KB. The construction of the inference engine was based on mathematical results presented in [11, 12], but its CoCoA[1] implementation was successively improved with substantial changes in the basic programs (not in the theory) to end in the implementation provided in this article.

The best known related work is Buchanan's and Shortliffe's classical work [17].

2.1 Some Basic Ideas about RBKS

Our KBs consist of "facts" (to be described in this section below) and logical formulae like:

$$\neg x[1] \wedge \neg x[2] \wedge x[3] \wedge \neg x[4] \rightarrow y[3]$$

These types of formulae are called "production rules". The above formula is read as follows:

⋆ IF not-$x[1]$ and not-$x[2]$ and $x[3]$ and not-$x[4]$ hold, THEN $y[3]$ holds.

The symbols of the form $x[1]$ and their negations, $\neg x[1]$, are called "literals".

Most production rules in this article have this form; nevertheless, in some cases, a prior grouping of subsets of rules makes the symbol "$\vee$" to appear intermixed with "$\wedge$".

The set of all literals that appear on the left-hand side of any production rule but not in any right-hand side, together with the set of their contrary literals (say, $x[1]$ is the contrary of $\neg x[1]$ and conversely) is called "the set of potential facts". The users of the RBKS can choose some subset of facts from the set of all potential facts (taking care not to choose simultaneously two contrary literals), in order to "fire" rules. A rule can be "fired" if all literals on its left-hand side are given as facts. The firing of the rule gives its right-hand side as output.

We shall briefly describe the construction of the KBs of the first and third subsystems and the the inference engines of the third subsystem.

[1] CoCoA, a system for doing Computations in Commutative Algebra. Authors: A. Capani, G. Niesi, L. Robbiano. Available via anonymous ftp from: `cocoa.dima.unige.it`

The inference engine works as follows: first, facts and production rules are automatically translated into polynomials and, second, these polynomials are treated under theoretical and computational constructs called Gröbner bases and Normal Forms, using the Computer Algebra language CoCoA [18, 19], in order to verify consistency and draw consequences.

2.2 Some Basic Logical and Mathematical Concepts

A logical formula A_0 is a "tautological consequence" of the formulae A_1,A_2,...,A_m if and only if whenever A_1, A_2, ..., A_m are true then A_0 is true.

Next, the expressions of the polynomials corresponding to the four basic bivalued logical formulae (those corresponding to the symbols $\neg$,$\vee$, $\wedge$ and $\rightarrow$) are provided. The uppercase letters represent the propositional variables that stand in the production rules and the lowercase letters represent their corresponding polynomial variables:

- $\neg X_1$ is translated into the polynomial $1 + x_1$
- $X_1 \vee X_2$ is translated into the polynomial $x_1 \cdot x_2 + x_1 + x_2$
- $X_1 \wedge X_2$ is translated into the polynomial $x_1 \cdot x_2$
- $X_1 \rightarrow X_2$ is translated into the polynomial $x_1 \cdot x_2 + x_1 + 1$.

These translations can be input almost directly in CoCoA (see Subsection 4.1). They allow to translate any logical formula into a polynomial; the coefficients of these polynomials are just 0 and 1, and the maximum power of variables is 1.

Theorem 1. *A formula A_0 is a tautological consequence of the formulae in the union of the two sets $\{A_1, A_2, ..., A_m\} \cup \{B_1, B_2, ..., B_k\}$ that represent, respectively, a subset of the set of potential facts and the set of all production rules of a RBKS if and only if the polynomial translation of the negation of A_0 belongs to the sum of the three ideals $I + K + J$ generated, respectively, by the polynomials $x_1^2 - x_1, x_2^2 - x_2, ..., x_n^2 - x_n$, by the polynomial translation of the negations of $A_1, A_2, ..., A_m$ and by the polynomial translation of the negations of $B_1, B_2, ..., B_k$.*

Explanatory Note 1: We are working in the quotient ring: $(Z/2Z)[x_1, x_2, ..., x_n]/I$, where I is as above. That justifies why the coefficients are just 0 and 1 and the maximum power of variables is 1. As CoCoA cannot perform calculations in residue class rings, instead of considering the ideal

$$K + J \subseteq (Z/2Z)[x_1, x_2, ..., x_n]/I$$

we shall consider

$$I + K + J \subseteq (Z/2Z)[x_1, x_2, ..., x_n] \ .$$

That A_0 is a consequence of the formulae in $\{A_1, A_2, ..., A_m\} \cup \{B_1, B_2, ..., B_k\}$, can be checked in CoCoA by typing:

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

"NF" means "Normal Form". If the output of the command is 0, the answer is "yes"; if the output is different from 0, the answer is "no". See Appendix II for details.

2.3 Application to the Study of Inconsistency

A RBKS is inconsistent if and only if any formula written in the language of the RBKS is tautological consequence of the formulae in the RBKS. In such case, contradiction is a consequence of the information contained in the RBKS.

Inconsistency is expressed by the algebraic fact that the element 1 of the ring of polynomials belongs to the ideal $I+K+J$ (I and J are always the same, K depends on the patient considered). The reason is that the ideal is the whole ring in this case, so that the theorem above would imply that all formulae are consequences of the RBKS, so that, particularly, contradictions are consequences of the RBKS (both if the underlying logic is Boolean or multi-valued).

This condition can be checked using Gröbner bases by typing the CoCoA command:

```
GBasis(I+J+K);
```

If the output is 1 (it appears as `[1]` on the screen) the RBKS is inconsistent; otherwise (it may be a large set of polynomials) the RBKS is consistent.

In the case of multi-valued logic, different types of consistency can be distinguished and interpreted from this point of view [20]:

– any formulae is a tautological consequence of a maximal consistent set of facts, the rules and the integrity constraints (week logic inconsistency)
– the conjunction of the facts in a maximal consistent set of facts, the rules and the integrity constraints can only take the truth value "false" (strong logic inconsistency).

3 Description of the First Subsystem: Possibility of Suffering the Illness

The first subsystem studies the influence of predisposing factors and triggering factors on the possibility of suffering migraine as a chronic illness. A literal is assigned to the existence or non-existence of each one of these factors.

3.1 Predisposing Factors

We consider the following factors as predisposition factors:

- Gender: woman, denoted as $x[1]$; man, denoted as $\neg x[1]$ (a relation $\frac{1}{2.3}$ man/woman exists among persons suffering migraine).
- Familiar antecedents: yes, denoted as $x[2]$; no, denoted as $\neg x[2]$ (this factor occurs in about 70% of the cases).
- Hormonal related predisposing factors: yes, $x[3]$; no, $\neg x[3]$.
- Age: $15 \leq age \leq 30$, $x[4]$; $age < 15$ or $age > 30$, $\neg x[4]$ (migraine is more frequent among people aged between 15 and 30).

3.2 Triggering Factors

Among the six factors treated by the system, we only mention the first and the fifth one.

- Dietetic triggering factors (abuse of food and drinks containing caffein, alcoholic drinks, especially red wine, vermouth, beer and champagne, milk and milk-based food, chemical additives, especially monosodic glutamate, some seasonings and spices, abuse of diets and irregularity in eating times): yes, $x[5]$; no, $\neg x[5]$.
- Hormonal related triggering factors (related to menstrual cycle, pregnancy, taking oral contraceptives): yes, $x[9]$; no, $\neg x[9]$.

We have actually made an over-simplification by placing all these factors at a same level with respect to time. This is because we only intend to build a prototype, capable to be improved in the future.

3.3 Production Rules Corresponding to Predisposing and Triggering Factors

The production rules for predisposing factors are constructed as follows. An empty Karnaugh diagram was built first (Table 1). A specialist in anorexia was asked to assign an intensity of predisposition to the illness ($y[1]$ = high intensity, $y[2]$ = medium intensity, $y[3]$ = low or null intensity) to each conjunction formed by an element of the first row and an element of the first column. For instance, the intensity $y[3]$ corresponds to the conjunction $\neg x[1] \wedge \neg x[2] \wedge x[3] \wedge \neg x[4]$. This way Table 1 was filled.

Each of these conjunctions implying its corresponding predisposition intensity constitutes a production rule. In the case of the conjunction above, this rule is:

	$x[3] \wedge x[4]$	$x[3] \wedge \neg x[4]$	$\neg x[3] \wedge \neg x[4]$	$\neg x[3] \wedge x[4]$
$x[1] \wedge x[2]$	$y[1]$	$y[2]$	$y[2]$	$y[1]$
$x[1] \wedge \neg x[2]$	$y[2]$	$y[2]$	$y[2]$	$y[2]$
$\neg x[1] \wedge \neg x[2]$	$y[2]$	$y[3]$	$y[3]$	$y[2]$
$\neg x[1] \wedge x[2]$	$y[2]$	$y[2]$	$y[2]$	$y[2]$

Table 1. Some of the production rules corresponding to predisposing and triggering factors

$$\neg x[1] \wedge \neg x[2] \wedge x[3] \wedge \neg x[4] \to y[3]$$

Well known simplification processes of logic allow to condense the rules to a set with fewer elements (five production rules, $R1$ to $R5$, in this case). For instance, the four production rules corresponding to the last row of the diagram are condensed into the production rule:

$$R5 : \neg x[1] \wedge x[2] \to y[2]$$

This rule represents the statement:

⋆ IF gender = man AND familiar antecedents = yes THEN intensity of predisposition to the illness = medium.

The construction of the set of rules corresponding to triggering factors is a little more complex because of the following two reasons. First, in each of the production rules corresponding to triggering factors, a predisposition intensity must occur in the antecedent of the rule, because the intensity of predisposition influences the possibility of suffering the illness when triggering factors occur. Second, the number of triggering factors is 6, while the number of predisposing factors was 4.

This gives rise to four Karnaugh diagrams of size 8×8 each, which are not included here.

Conjunctions of triggering factors and negations of triggering factors, together with their corresponding predisposition intensities are assigned a possibility level (the possibility of suffering the illness). The propositional variables $w[1], w[2]$ and $w[3]$ represent "the possibility of suffering migraine is high/medium/low or null", respectively.

The simplification of all combinations that result from the mentioned Karnaugh diagrams gives rise to 70 production rules. For instance R6 and R40 are:

$$R6 : y[1] \wedge x[5] \wedge x[6] \wedge x[7] \to w[1]$$
$$R40 : y[2] \wedge x[5] \wedge \neg x[6] \wedge x[7] \wedge x[8] \wedge (x[9] \vee x[10]) \to w[1]$$

R40, for instance, represents,

⋆ IF intensity of predisposing factors = medium AND (influence of) dietetic factors = yes AND (influence of) ambient factors = no AND (influence of) emotional factors = yes AND ((influence of) hormonal factors = yes) OR (other triggering factors = yes), THEN possibility of suffering migraine = high.

4 Third Subsystem: Treatment

When a patient has been diagnosed chronic migraine, specialists may suggest first a change in the style of life or elimination, when possible, of triggering factors. In addition, depending on the frequency and intensity of the episodes (propositional variables $x[1]$ and $x[2]$ below), different types of medicines are usually recommended (we have classified them in three generic types $t[1]$, $t[2]$, $t[3]$).

Some counter-indications are dealt with later: self-medication and medication abuse is frequent among persons suffering migraine.

The production rules in this third subsystem are based on the information provided in [1–4] and on the advice of experts.

- Frequency of episodes: occasional, $x[1]$; frequent, $\neg x[1]$.
- Intensity of episode[2]: low/moderate, $x[2]$; moderate/severe, $x[3]$.
- Medicines of type I (mostly analgesics and/or anti-inflammatories): yes, $t[1]$; no, $\neg t[1]$.
- Medicines of type II (tryptans or ergotamines): yes, $t[2]$; no = $\neg t[2]$.
- Medicines of type III (prophylactic treatment, as treatment with β-blockers): yes, $t[3]$; no, $\neg t[3]$.

Combining (some of) these literals, the following three production rules represent a classification of the generic type of medicines which can be recommended to a patient.

$$R1 : x[1] \wedge x[2] \rightarrow t[1]$$
$$R2 : x[1] \wedge x[3] \rightarrow t[2]$$
$$R3 : \neg x[1] \wedge x[3]. \rightarrow t[3]$$

R2, for instance, says:

⋆ IF frequency of episodes: occasional = yes AND intensity of episode: moderate/severe = yes, THEN medicines of type II = yes.

Facts $x[1], x[2], x[3]$ have been already treated. Next 13 factors are considered in the system, represented by literals $x[i]$ or $\neg x[i]$ $(i = 4, ..., 16)$. They express counter-indications to some medications. For the sake of space we only refer to the first four ones.

- Asthma: yes, $x[4]$; no = $\neg x[4]$.
- Congestive hearth insufficiency: yes, $x[5]$; no, $\neg x[5]$.
- Diabetes: yes, $x[6]$; no, $\neg x[6]$.
- Peripheral arteriopathy: yes, $x[7]$; no, $\neg x[7]$.

The different possible combinations of the literals $x[i]$ or $\neg x[i]$ $(i = 4, ..., 16)$ with the literals $t[j]$ or $\neg t[j]$ $(j = 1, ..., 3)$, imply one of the following prophylactic treatments.

- Prophylactic treatment with β-blockers: yes, $y[1]$; no. $\neg y[1]$.
- Prophylactic treatment with anti-convulsives: yes, $y[2]$; no, $\neg y[2]$.
- Prophylactic treatment with anti-depressives: yes, $y[3]$; no, $\neg y[3]$.
- Prophylactic treatment with blockers of channels of calcium: yes, $y[4]$; no, $\neg y[4]$.
- Prophylactic treatment with agonists of serotonine re-captation: yes, $y[5]$; no, $\neg y[5]$.

The implications between combinations of the $x[i]$, $\neg x[i]$ $(i = 4, ..., 16)$ with $t[j]$, $\neg t[j]$ $(j = 1, ..., 3)$ in the antecedents and $y[k]$, $\neg y[k]$ (k=1,..,5) in the consequents, give rise to the following production rules, which refer to the prophylactic treatment with β-blockers.

$$R4 : t[3] \wedge \neg x[4] \wedge \neg x[5] \wedge \neg x[6] \wedge \neg x[7] \rightarrow y[1]$$
$$R5 : \neg t[3] \vee x[4] \vee x[5] \vee x[6] \vee x[7] \rightarrow \neg y[1]$$

For instance, R4 represents:

⋆ IF medicines of type III (prophylactic treatment) = yes AND asthma = no AND congestive hearth insufficiency = no AND diabetes = no AND peripheral arteriopathy = no THEN prophylactic treatment with β-blockers = yes.

Production rules R6 and R7 refer to the prophylactic treatment with anti-convulsives.

$$R6 : t[3] \wedge \neg x[8] \rightarrow y[2]$$
$$R7 : \neg t[3] \rightarrow \neg y[2]$$

For instance, R6 represents:

[2] A literal and its negation haven't been used in this case because the two possibilities are not exclusive.

⋆ IF medicines of type III (prophylactic treatment) = yes AND altered hepatic function = no, THEN prophylactic treatment with anti-convulsives = yes

Only two among all the production rules referring to the prophylactic treatments with anti-depressives, blockers of calcium channels and agonists of serotonine re-captation are transcribed next. Their meaning, similar to the ones just given for R4 and R6, is straightforward.

$$R8 : t[3] \wedge \neg x[6] \wedge \neg x[8] \wedge \neg x[9] \wedge \neg x[10] \wedge \neg x[11] \wedge \neg x[12] \rightarrow y[3]$$
$$R13 : \neg t[3] \vee x[8] \vee x[16] \rightarrow \neg y[5]$$

The next two production rules state a relation of a partial incompatibility of $t[1]$ and $t[2]$ with $t[3]$. They must be understood as, first, if it is enough to treat a patient's migraine with medicines of type I or II, do not prescribe medicines of type III and, second, if the patient does not react positively to medicines of type I and II, he/she should be treated with medicines of type III. For the sake of simplicity we do not include any statement that would translate the possibility of being treated with medicines of type III and, at the same time, being recommended to use medicines of both or one of the two other types.

$$R14 : t[1] \vee t[2] \rightarrow \neg t[3]$$
$$R15 : \neg t[1] \wedge \neg t[2] \rightarrow t[3]$$

4.1 CoCoA Implementation of the Inference Engine of the Third Subsystem

First steps

The commands are written in ``typewriter'' font, while the explanations are written in normal font.

The polynomial ring A with coefficients in $Z/2Z$ (that is, allowing coefficients 0 and 1), 16 variables x, three variables t and five variables y and the ideal I are declared as follows.

```
A::=Z/(2)[x[1..16],t[1..3],y[1..5]];
USE A;

I:=Ideal(x[1]^2-x[1],...,x[16]^2-x[16],t[1]^2-t[1],...,t[3]^2-t[3],y[1]^2-y[1],
         ...,y[4]^2-y[4],y[5]^2-y[5]);
```

(note that "`..`" is an abbreviation accepted by CoCoA, unlike "`,...,`" that is used here to save space and is not acceptable code).

The following commands (see Section 2.2 above) produce the polynomial translation of bivalued logical formulae. CoCoA does not admit the definition of infix operators, so that it requires that logical formulae be written in prefix form. `NEG`, `OR1`, `AND1`, `IMP` will denote $\neg$, $\vee$, $\wedge$, $\rightarrow$, respectively.

```
NEG(M):=NF(1+M,I);
OR1(M,N):=NF(M+N+M*N,I);
AND1(M,N):=NF(M*N,I);
IMP(M,N):=NF(1+M+M*N,I);
```

Entering the Rule-Based Knowledge System

All 15 production rules of the third subsystem should be entered first. As said above, CoCoA requires that formulae are written in prefix form. For instance,

$$R1 : x[1] \wedge x[2] \rightarrow t[1]$$

is rewritten as:

```
R1:=NF(IMP(AND1(x[1],x[2]),t[1]),I);
```

The set of potential facts should be entered. Each patient is characterized by the factors and symptoms that form a subset of the whole set of potential facts, subject to the condition that such a subset contains one and only one element of each pair formed by a fact `Fi` and its contrary `FiN` ($i \in \{1, 4, ..., 16\}$); and either `F2` or `F3`.

```
F1:=x[1];              F1N:=NEG(x[1]);
F2:=x[2];
F3:=x[3];
F2:=x[4];              F1N:=NEG(x[4]);
F2:=x[5];              F1N:=NEG(x[5]);
   ...                     ...
   ...                     ...
F16:=x[16];            F16N:=NEG(x[16]);
```

The ideal J, generated by the 15 production rules of the systems is:

```
J:=Ideal(NEG(R1),NEG(R2),NEG(R3),NEG(R4),NEG(R5),NEG(R6),NEG(R7),NEG(R8),NEG(R9),
         NEG(R10),NEG(R11),NEG(R12),NEG(R13),NEG(R14),NEG(R15));
```

(recall that Theorem 1 implies the need of entering `NEG` before the rules; the same holds for the ideal K below).

Let us consider, as illustration, the following ideal K that characterizes a patient by factors $\neg x[1]$, $x[3]$, $x[4]$, $x[5]$, $\neg x[6]$, $x[7]$, $\neg x[8]$, $\neg x[9]$, $\neg x[10]$, $\neg x[11]$, $\neg x[12]$, $x[13]$, $x[14]$, $x[15]$, $\neg x[16]$.

```
K:=Ideal(NEG(F1N),NEG(F3),NEG(F4),NEG(F5),NEG(F6N),NEG(F7),NEG(F8N),NEG(F9N),
         NEG(F10N),NEG(F11N),NEG(F12N),NEG(F13),NEG(F14),NEG(F15),NEG(F16N));
```

Checking for Consistency

Once the whole set of rules and potential facts has been written down, it is necessary to check its consistency. Recall that consistency is checked by using the command `GBasis`.

Checking for consistency helps not only to suppress inconsistencies, but also to improve the KB (always interacting with the experts).

In the example above, typing:

```
GBasis(I+K+J);
```

no inconsistency was found (`[1]` was not returned). In the way the set of production rules has been built here, with only one literal as consequent, it is not probable to find inconsistencies other that those resulting from misprints.

Explanatory Note 2: Observe that the complete consistency checking of this RBKS would need to check that each possible (maximal consistent) set of facts together with the rules does not produce any inconsistency. As this is a huge task (there are $2^{15} = 32768$ maximal sets in this case), we have decided to check the consistency for the data of each patient before applying the knowledge extraction for him/her. So the consistency is checked for the information contained in $I + K + J$ (for different ideals K). If any inconsistency was found it should be reported to the authors for debugging.

Explanatory Note 3: Nevertheless, other types of inconsistencies may occur due to what are known as “integrity constraints”, denoted “IC”. An integrity constraint is a conjunction of literals which the experts judge that can never hold simultaneously.

For instance, an IC can be the conjunction $\neg\alpha \wedge \beta$ of two literals. Thus, the negation “NIC” of the integrity constraint, in our example $\neg(\neg\alpha \wedge \beta)$, ought to be added to the expert system as new information.

If by firing some rules from the given facts, one finds all the literals in the IC, in our case both $\neg\alpha$ and β, an inconsistency takes place, in our case $\neg\alpha \wedge \beta$ and $\neg(\neg\alpha \wedge \beta)$. For example, if, instead of including in medicines of type II, both tryptans and ergotamines, the third subsystem separated them into two types, an IC “tryptans AND ergots” should be introduced, because they are incompatible (as simultaneous treatments).

Extraction of Consequences

Let us consider, as illustration, the ideal K defined above. Let us ask if any of the prophylactic treatments $y[i]$ $(i = 1, ..., 5)$ should be recommended to the patient characterized by the ideal K. The following commands:

```
NF(NEG(y[1]),I+K+J);
NF(NEG(y[2]),I+K+J);
NF(NEG(y[3]),I+K+J);
NF(NEG(y[4]),I+K+J);
NF(NEG(y[5]),I+K+J);
```

give, as output, 1, 0, 0, 1 and 0, respectively. It means that any of the prophylactic treatments $y[2]$, $y[3]$ and $y[5]$ could be recommended.

As a matter of fact, as the system does not substitute the expert, who must have the last word about treatment.

5 Considerations on the Use of a Gröbner Bases Approach

In [21], a propositional Gröbner proof system is discussed. It is shown that this system polynomially simulates Horn clause resolution and weakly exponentially simulates resolution. The authors say that this suggests that the Gröbner bases algorithm might replace resolution as a basis for heuristics for NP-complete problems (Kapur and Narendran already stated in [8] that using a Gröbner bases approach subsumes resolution). Let us observe that there is an important difference between the average case and worst case complexities of Gröbner bases computations. In [21] there is also a comparison with other methods, which result only slightly superior. Thus, from the point of view of complexity, using a Gröbner bases approach seems reasonable.

This was also the opinion of the referee of [11], an expert in automated theorem proving, who judged our approach as competitive even though not the best ([11], page 8).

Our choice is mainly based on our previous experience in developing medium size RBKSs and other related simulations. We have developed the theoretical aspects of an algebraic approach to the verification and knowledge extraction in RBKS with Boolean or multi-valued modal underlying logic, that uses Gröbner bases [11, 12]. We have used it to study:

- verification and decision taking in medical appropriateness criteria for revascularization in case of coronary deseases [13] (it uses three-valued and modal logic)
- early detection, diagnoses and treatment of anorexia [14] (it uses Boolean propositional logic)
- ...

We have also developed a Gröbner bases approach to topology-independent railway interlocking systems [15]. We had studied the problem both using Prolog and with a matrix approach [22], obtaining slightly worse although comparable timings. Nevertheless, the order of magnitude of the problems that can be treated is the same.

In fact, after developing a passengers flow simulation package for the Spanish Airport Authority (AENA), we are now developing a Gröbner bases based model for Advanced Surface Movement Guidance and Control Systems (A-SMGCS).

An advantage of this approach is the possibility of reusing the Gröbner basis corresponding to the part of the KB that contains the set of production rules. Therefore, an update does not require to start the computation from the beginning.

There are two more advantages in our approach:

- it can deal with multi-valued logics
- logical formulae can be written without any restrictions (they do not have to be Horn clauses; $\vee$ symbols can appear in the consequent).

A future extension made possible by this approach is the application of the method of hypotheses completion used in automatic discovery in Geometry [23–25] to knowledge completion in RBKSs.

Summarizing, our approach seems to be competitive when dealing with problems of small and medium size (e.g. RBKS with 80 variables and 120 rules, railway stations with 25 or 40 sections...) or in bigger problems that can be divided into subsystems.

6 Conclusions

We have presented a prototype of a RBKS for the study of migraine. At the present state, the RBKS is able to produce automatically a diagnosis of the illness. In addition, the system evaluates the different treatments of migraine.

Obviously the goal is not to substitute the specialist. There are two possible uses: first, to help in the detection of the illness by non-specialists (that must send the patient to a specialist!) and, second, to allow the specialist to compare his diagnosis and prescription with that one suggested by the system.

Its KB could be improved, detailed or updated without modifying its inference engine and knowledge treatment, as described in the paper.

Acknowledgements

This work is partially supported by the research projects TIC2000-1368-C03-01 and TIC2000-1368-C03-03 (Ministry of Science and Technology, Spain).

We would like to express our gratitude to the referees of this article, whose most valuable comments and suggestions have made possible to improve it considerably.

References

1. T. B. Smith, La prevención de la migraña ha dejado de ser un quebradero de cabeza, Tiempos Médicos 560 (1999) 23-29.
2. URL: http://www.ama-assn.org/special/migraine/migraine.htm
3. URL: http://www.amerifarma.com
4. URL: http://www.iladiba.com
5. W.W. Adams, P. Loustaunau, An Introduction to Gröbner Bases (Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 1994).
6. T. Becker, V. Weisspfenning, Gröbner bases (Springer, New York, 1993).
7. F. Winkler, Polynomial Algorithms in Computer Algebra (Springer-Verlag, Vienna, 1996).
8. D. Kapur, P. Narendran, An Equational Approach to Theorem Proving in First-Order Predicate Calculus (General Electric Corporate Research and Development Report 84CRD296, Schenectady, NY, March 1984, rev. December 1984). Also in: Proceedings of IJCAI-85 (1985) 1146-1153.
9. J. Hsiang, Refutational Theorem Proving using Term-Rewriting Systems, Artif. Intell. 25 (1985) 255-300.
10. J. Chazarain, A. Riscos, J.A. Alonso, E. Briales, Multivalued Logic and Gröbner Bases with Applications to Modal Logic, J. Symb. Comp. 11 (1991) 181-194.
11. L.M. Laita, E. Roanes-Lozano, L. de Ledesma, J. A. Alonso, A Computer Algebra Approach to Verification and Deduction in Many-Valued Knowledge Systems, Soft Comp. 3/1 (1999) 7-19.
12. E. Roanes-Lozano, L.M. Laita, E. Roanes-Macías, A Polynomial Model for Multivalued Logics with a Touch of Algebraic Geometry and Computer Algebra, Math. Comp. Simul. 45/1 (1998) 83-99.
13. 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, Comp. Math. Appl. 42/12 (2001) 1505-1522.
14. C. Pérez. L.M. Laita, E. Roanes-Lozano, L. Lázaro, J. González, L. Laita, A Logic and Computer Algebra-Based Expert System for Diagnosis of Anorexia, Math. Comp. Simul. 58 (2002) 183-202.
15. E. Roanes-Lozano, E. Roanes-Macías, Luis M. Laita, Railway Interlocking Systems and Gröbner bases, Math. Comp. Simul. 51 (2000) 473-481.
16. E. Mendelson, Boolean Algebra and Switching Circuits (Schaum, McGraw-Hill, New York, 1970).

17. B. Buchanan, E.H. Shortliffe, Editors, Rule Based Expert Systems: the MYCIN Experiments of the Stanford Heuristic Programming Project (Addison Wesley, New York, 1984).
18. A. Capani, G. Niesi, CoCoA User's Manual v. 3.0b (Dept. of Mathematics, University of Genova, Genova, 1996).
19. D. Perkinson, CoCoA 4.0 Online Help-electronic file acompanying CoCoA v.4.0 (2000).
20. E. Roanes-Lozano, E. Roanes-Macías and L.M. Laita, Geometric Interpretation of Strong Consistency of Knowledge Based Systems, Procs. of CASC'99 (Springer-Verlag, Berlin-Heidelberg, 1999) 349-364.
21. M. Clegg , J. Edmonds , R. Impagliazzo, Using the Groebner basis algorithm to find proofs of unsatisfiability, Procs. of the twenty-eighth annual ACM Symp. on Theory of Computing (Philadelphia, Pennsylvania, 1996) 174-183.
22. E. Roanes-Lozano, Luis M. Laita, An applicable topology-independent model for railway interlocking systems, Math. Comp. Simul. 45 (1998) 175-183.
23. D. Kapur, J.L. Mundy, Wu's method and its application to perspective viewing, in: Geometric Reasoning (MIT Press, 1989).
24. T. Recio, M.P. Vélez, Automatic Discovery of Theorems in Elementary Geometry, J. Aut. Reas. 23 (1999) 63-82.
25. E. Roanes-Macías, E. Roanes-Lozano, Automatic Determination of Geometric Loci. 3D-Extension of Simson-Steiner Theorem, in: Artificial Intelligence and Symbolic Computation, Procs. of AISC'2000 (Springer-Verlag, LNCS 1930, Berlin-Heidelberg, 2001) 157-173.

7 Appendix I: Example of Screenshot of the GUI

The screenshot in Fig. 1, refers to triggering factors. The user selects the keys corresponding to the patient, that are translated into literals. The GUI is in Spanish, but the English terms are similar.

8 Appendix II: Some Notes on the Theoretical Background of the Inference Engine

Although the underlying logic in the RBKS described in the article is Boolean logic, we shall detail here the general case of any p-valued logic (p prime).

We use the following restricted notion of propositional logic. It consists of

(i) a set $X = \{X_1, X_2, ..., X_n\}$ of propositional variables and a set $C = \{c_1, c_2, ..., c_t\}$ of connectives, from which well formed propositional formulae, denoted by the letter A (with or without subscripts) are constructed. These formulae form another set denoted $P_C(X_1, X_2, ..., X_n)$.

(ii) a set $L = \{0, 1, ..., p-1\}$ (where p is a prime number). The elements $0, 1, ..., p-1$ are considered as the truth values of the p-valued logic. The number $p-1$ represents the value "true", 0 represents "false", and the other elements represent intermediate truth values.

(iii) for each connective $c_j \in C$, a set of truth tables defined by functions $H_j : L^{s_j} \longrightarrow L$ (s_j is the arity of the connective c_j). The expression $c_j(A_1, ..., A_{s_j})$ represents the formula constructed by applying the connective c_j to the formulae $A_1, ..., A_{s_j}$.

(iv) a function v called "valuation" $v : X \longrightarrow L$.

(v) for each v another function $v' : P_C(X_1, X_2, ..., X_n) \longrightarrow L$, recursively defined as follows:
$v'(A) = v(A)$ if $A \in X$
$v'(A) = H_j(v'(A_1), ..., v'(A_{s_j}))$ if A is well formed from c_j and $A_1, ..., A_{s_j}$.

(vi) a relation named "tautological consequence": given A_0 and $A_1, A_2, ..., A_m$ in $P_C(X_1, X_2, ..., X_n)$, A_0 is a tautological consequence of $A_1, A_2, ..., A_m$, whenever if $v'(A_1) = p-1$, $v'(A_2) = p-1$,..., $v'(A_m) = p-1$ then $v'(A_0) = p-1$. Intuitively, a formula A_0 is a tautological consequence of other formulae $A_1, A_2, ..., A_m$ if and only if whenever $A_1, A_2, ..., A_m$ are true, A_0 is also true. $\{A_1, A_2, ..., A_m\} \models A_0$ denotes that A_0 is a tautological consequence of $A_1, A_2, ..., A_m$.

Let us assign a polynomial to each logical formula. This is achieved by assigning to each propositional variable X_i a monomial x_i and defining, for each connective c_j, a function:

$$f_j : ((Z/pZ)[x_1, x_2, ..., x_n]/I)^{s_j} \longrightarrow (Z/pZ)[x_1, x_2, ..., x_n]/I \ .$$

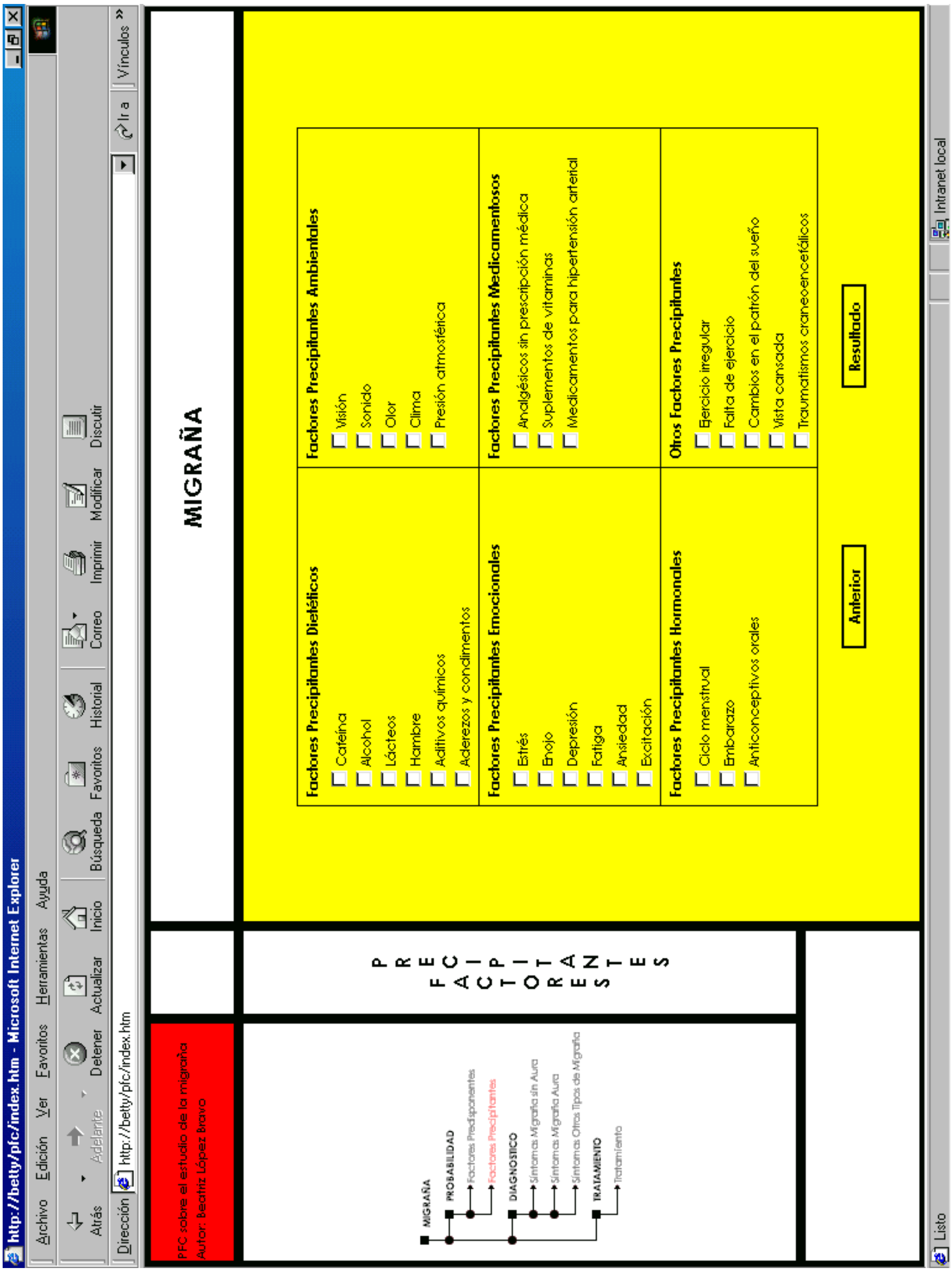

Fig. 1. A screenshot of the GUI

The symbol I represents the ideal generated by the polynomials $x_1^p - x_1, x_2^p - x_2, ..., x_n^p - x_n$:

$$I = \langle x_1^p - x_1, x_2^p - x_2, ..., x_n^p - x_n \rangle$$

As the process has been published by the authors elsewhere [11, 12], we simply transcribe as illustration the final expressions for the function $f_{\rightarrow}$ for Boolean logic:

$$f_{\rightarrow}(q, r) = (1 + q + q \cdot r) + I$$

and for Kleene's three-valued and modal logic:

$$f_{\rightarrow}(q, r) = (q^2 \cdot r^2 + q^2 \cdot r + q \cdot r^2 + 2 \cdot q + 2) + I$$

(the complexity of the polinomial expressions increase with the value of p).

The functions f_j translate the basic propositional formulae $\neg X_i, X_i \wedge X_k, X_i \vee X_k, X_i \rightarrow X_k$ (and $\Diamond X_i, \Box X_i$ in the case of three-valued modal logic) into (classes of) polynomials. The next definition determines a function θ that, interacting with the functions f_j, translates any propositional formula, in particular the rules and other items of any RBS, into (classes of) polynomials.

$$\theta : P_C(X_1, X_2, ..., X_n) \longrightarrow (Z/pZ)[x_1, x_2, ..., x_n]/I$$

is a function from propositions to (classes of) polynomials, recursively defined as follows:

$$\theta(X_i) = x_i + I, \text{ for all } i = 1, ..., n$$
$$\theta(A) = f_j(\theta(A_1), ..., \theta(A_{s_j})) \text{ if } A \text{ is } c_j(A_1, ..., A_{s_j}) \ .$$

For each valuation v, v^* is the homomorphism:

$$v^* : (Z/pZ)[x_1, x_2, ..., x_n]/I \longrightarrow Z/pZ$$

such that $v^*(x_i + I) = v(X_i)$ for $i = 0, ..., n$.

It can be proved that for any valuation v, $v' = v^* \cdot \theta$.

Lemma 1. $\{0\} + I = (\bigcap_{i=1,..,k} \ker(v_i^*)) \cap (\theta(P_C(X_1, X_2, ..., X_n)))$, *where* $k = p^n$ *is the number of all valuations, the* $v_i's$ *range over all possible valuations.*

Lemma 2. *Let* $A_1, A_{2,...,}A_m, A_0 \in P_C(X_1, X_2, ..., X_n)$. *The following two assertions are equivalent:*

(i) for all valuations $v_i (i = 1, ..., k)$ *such that* $v_i^*(\theta(A_1)) = v_i^*(\theta(A_2)) = ... = v_i^*(\theta(A_m)) = 0$ *it follows that* $v_i^*(\theta(A_0)) = 0$.

(ii) $\theta(A_0) \in \langle \theta(A_1), \theta(A_2), ..., \theta(A_m) \rangle$.

Theorem 1. (revisited) *Let* $A_0, A_1, ..., A_m \in P_C(X_1, X_2, ..., X_n)$. *The following assertions are equivalent:*

(i) $\{A_1, A_2, ..., A_m\} \models A_0$,

(ii) $f_{\neg}(\theta(A_0)) \in \langle f_{\neg}(\theta(A_1)), ..., f_{\neg}(\theta((A_m)) \rangle$.

Proof.- $\{A_1, A_2, ..., A_m\} \models A_0$ iff for any v, $v'(A_1) = p - 1, ..., v'(A_m) = p - 1$ implies $v'(A_0) = p - 1$ (remember that $p - 1$ is the value "true"). This is equivalent to the condition: $(p - 1) - v'(A_1) = 0, ..., (p - 1) - v'(A_m) = 0$ implies $(p - 1) - v'(A_0) = 0$ which is equivalent to $v'(\neg A_1) = 0, ..., v'(\neg A_m) = 0$ implies $v'(\neg A_0) = 0$.

This implication is equivalent to that, for any v,

$$v_i^*(\theta(\neg A_1)) = 0, ..., v_i^*(\theta(\neg A_m)) = 0 \text{ implies } v_i^*(\theta(\neg A_0)) = 0$$

which is equivalent to

$$v_i^*(f_{\neg}(\theta(A_1))) = 0, ..., v_i^*(f_{\neg}(\theta(A_m))) = 0 \text{ implies } v_i^*(f_{\neg}(\theta(A_0)) = 0 \ .$$

By the Lemma 2 above, the last implication is equivalent to

$$f_{\neg}(\theta(A_0)) \in \langle f_{\neg}(\theta(A_1)), ..., f_{\neg}(\theta(A_m)) \rangle \ .$$