Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(2): 2938–2963.
DOI: 10.3934/math.2024145
Received: 20 October 2023
Revised: 19 December 2023
Accepted: 27 December 2023
Published: 02 January 2024

Research article

A novel way to build expert systems with infinite-valued attributes

Antonio Hernando[1], **José Luis Galán-García**[2,*] **and Gabriel Aguilera-Venegas**[2]

[1] Depto. de Sistemas Informáticos, E.T.S.I. de Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid, Spain

[2] Departamento de Matemática Aplicada, Escuela de Ingenierías Industriales, Universidad de Málaga, Málaga, Spain

* **Correspondence:** Email: jlgalan@uma.es.

Abstract: An expert system is a computer program that uses the knowledge of an expert to solve problems in a specific domain. Expert systems are used in a wide variety of fields, such as medicine, financial diagnosis and engineering. The attributes of an expert system are the characteristics of the problems that the system can solve. In traditional expert systems, attributes typically have a finite number of possible values. However, in scenarios where an attribute can assume a value from an infinite (or significantly large finite) set, the expert system cannot be represented using propositional logic. Until now, no method had been identified to implement such a system on a Computer Algebra System. Here, we break new ground by presenting a model that not only addresses this gap but also provides a fresh perspective on previous results. In fact, these prior results can be viewed as specific instances within the broader framework of our proposed solution. In this paper, we put forth an algebraic approach for the development of expert systems capable of handling attributes with infinite values, thereby expanding the problem-solving capacity of these systems.

Keywords: Gröbner bases; logic and symbolic computing; rule based expert systems
Mathematics Subject Classification:

1. Introduction

Expert systems are computational programs designed to emulate the decision-making process of human experts within a specific field. One effective method for representing knowledge in such a system is through propositional logic, where knowledge inference is intrinsically linked to the concept of Tautological consequence. Utilizing a mathematical result [1] based on prior work [2–6], this issue can be converted into an algebraic problem involving the calculation of certain Gröbner bases [7, 8]. Consequently, expert systems predicated on propositional logic can be readily implemented using a computer algebra system like CoCoA [9]. This approach has facilitated the development of various expert systems in recent years [10–19].

The 'Concept-Attribute-Value' paradigm provides an alternative method for representing knowledge in Expert Systems. This approach is often more natural and efficient, offering several advantages over propositional logic. When the possible values of attributes can only take a limited set of values, knowledge inference under this paradigm can be translated into algebraic terms [20]. This translation facilitates the implementation of an Expert System based on the 'Concept-Attribute-Value' paradigm within Computer Algebra Systems. Despite the differences between the algebraic approach based on Boolean propositional logic and the 'Concept-Attribute-Value' paradigm, they are in some ways equivalent. Specifically, any Expert System that can be implemented in one model can also be implemented in the other, provided that attributes can assume only a finite set of values [21].

However, there are instances where it is not feasible to impose this limitation, necessitating the consideration of scenarios where an attribute, x, can assume one value from an infinite (or very large finite) set. In such cases, we may encounter formulae like ($x \neq 2$ or $x \neq 0 \rightarrow y = 0$). In these circumstances, the expert system cannot be represented using propositional logic, as x can take on an infinite number of possible values. Up until now, no method had been found to implement it on a Computer Algebra System. However, we have made a breakthrough and will present our newly discovered method in this paper.

In this paper, we address these specific constraints and introduce an innovative algebraic methodology for the development of expert systems. This methodology takes into account variables or attributes that are capable of assuming a value from an infinite set. As we will demonstrate, this model offers a fresh perspective on previous results, to the point where they can be viewed as specific instances of our proposed solution.

The structure of this paper is outlined as follows: In Section , we conduct a comparative analysis of our models with related ones. Section 3 introduces the representation formalism used to implement expert systems, wherein an attribute can take on a value from an infinite set. In Section 4, we provide proofs related to the proposed algebraic model for implementing this type of expert system. Section 5 provides a deeper understanding of our proposed algebraic model. This model is illustrated using a real-world example: A railway interlocking system that can identify hazardous situations in a railway station (see Section 6 for details). Finally, in Section 7, we summarize our conclusions and discuss the potential implications of our findings.

2. Comparison with related works

Our approach is centered on harnessing the capabilities of Computer Algebra Systems for the swift and effective development of expert systems. These systems represent knowledge, input and output through polynomials on multiple variables. By employing a representation paradigm based on propositional logic or 'Concept-Attribute-Value', we can translate knowledge using these polynomials. These algebraic models frame the problem of determining the system's output as an algebraic problem, typically the ideal membership problem, which can be resolved using Gröbner bases. In instances where we utilize propositional logic that accommodates uncertainty, we can deploy expert systems that handle these situations, making it particularly suitable in fields like medicine.

However, all these algebraic models are constrained by the assumption that the values of the variables, or attributes, necessary for knowledge representation must be confined to a potentially finite set (in the case of propositional logic, all variables are Boolean, with possible values True, False).

The novelty of our paper lies in its provision of an algebraic model that accommodates variables not restricted to a potentially finite set of values, allowing for an infinite range of potential variable values. In such cases, our model, unlike its predecessors, can implement these systems with Computer Algebra Systems using the proposed algebraic model. In essence, our approach extends previous models.

We believe our strategy can be applied to various expert systems where uncertainty is not a factor. It proves especially effective for decision trees that culminate in a finite set of outputs rather than intermediate steps and results with fluctuating levels of certainty.

However, our model does present some limitations:

- Our model does not account for uncertainty in knowledge. We plan to extend our algebraic model to consider uncertainty in future work.
- Our strategy has some limitations when variables take on a value from an infinite set. We have only considered relations between variables with equality ($x_i = x_j$) and inequality ($x_i \neq x_j$). The ability to represent order relations such as $x_i > x_j$ is currently beyond our scope.

Similar to preceding algebraic models, our model relies on the computation of Gröbner bases, thus presenting the same degree of complexity, and the inference engine operates within comparable time frames. However, in instances where all variables are Boolean, we can employ computer algebra systems like Polybori [22], which are tailored for Boolean polynomials, leading to significantly more efficient systems.

Despite these limitations, our strategy provides a framework that can be readily applied in scenarios where systems do not incorporate uncertainty. In this paper, we have demonstrated a practical application of our strategy in addressing interlocking problems (see Section 6).

Like previous algebraic models, our approach implements an expert system through a Computer Algebra System. While our approach is of theoretical interest, it shares a practical limitation with previous models. Since Computer Algebra Systems have not been certified for use in safety-critical implementations, systems developed with our framework cannot be integrated with safety instrumented system computers to achieve the targeted Safety Integrity Level (SIL). However, the results may prove beneficial for simulations that do not require certification credit.

3. Expert Systems

In this section, we will explore the fundamental concepts and principles that underpin the algebraic methodologies employed in the development of Expert Systems within Computer Algebra Systems. Expert Systems are computational programs characterized by three core components:

Input. The input of the expert system is the collection of facts observed in the environment. The set is denoted by the symbol $\mathcal{F}$.

Knowledge-Base. The knowledge-base encapsulates the information stored within the system. This information, used in conjunction with the input of the expert system, is used to inference the output of the system. The knowledge, which mirrors that of an expert, is expressed as a finite set of formulae $\mathcal{K}$.

Output. The output of the classification expert system constitutes the knowledge inferred from the input and the knowledge base within the expert system.

Each of the three components within an expert system necessitates the representation of knowledge. Analogous to Propositional Logic, we posit that knowledge in an expert system is depicted through a finite set of variables $x_1 \dots x_N$. However, diverging from propositional logic, we do not confine variables to Boolean values; in fact, variables may assume a value from an infinite set of values. In the following definition, we will formally establish the conceptual framework of an expert system

Definition 3.1 (Conceptual Framework)**.** *A conceptual ground is* $(\mathcal{X}, \mathcal{V}, \Psi)$ *where* $\mathcal{X}$ *is a finite set of possible 'variables',* $\mathcal{V}$ *is a (non-necessary) finite set of possible 'values' and* Ψ *is a function* $\mathcal{X} \longrightarrow \mathcal{P}(\mathcal{V})$ *where* $\mathcal{P}(\mathcal{V})$ *represents the power set of* $\mathcal{V}$. $\Psi(x)$ *represents the possible values that the variable* x *may take.*

The aforementioned definition encapsulates the expert system predicated on Propositional Logic. In the context of Propositional Logic, as per Definition 3.1, we have $\mathcal{X} = x_1 \dots x_N$, $\mathcal{V} = \{True, False\}$ and each variable x_i assumes the potential values $\{True, False\}$, that is to say, $\Psi(x_i) = \{True, False\}$.

The information of the environment is represented by means of states. A state is an instantiation of the conceptual framework: every variable x_i takes a value from the set of its potential values $\Psi(x_i)$. Formally:

Definition 3.2 (State)**.** *A state* S *is defined as a function* $S : \mathcal{X} \longrightarrow \mathcal{V}$. *We designate* $\mathcal{S}$ *as the set encompassing all states.*

Given a state, $s \in \mathcal{S}$, we can state relations between the variables $\mathcal{X}$ by means of formulae (in the same way, as formulae in propositional logic relates boolean variables).

Definition 3.3 (Formula)**.** *A formula is defined as follows:*

- *Positive Atomic Formula.*
  - $x = v$, *where* $x \in \mathcal{X}$ *is a variable and* $v \in \Psi(x)$ *is a possible value of* x.
  - $x = y$, *where* $x, y \in \mathcal{X}$ *are variables.*
- *Negative Atomic Formula.*
  - $x \neq v$, *where* $x \in \mathcal{X}$ *is a variable and* $v \in \Psi(x)$ *is a possible value of* x.
  - $x \neq y$, *where* $x, y \in \mathcal{X}$ *are variables.*
- *Disjunctive of atomic formulae:*

$$A_1 \vee ... \vee A_r$$

  where $A_1, ..., A_r$ *are positive or negative atomic formulae.*

We designate C *as the set encompassing all formulae.*

Definition 3.4. *Given an atomic formula* A*, we will denote the atomic formula* $\neg A$ *as the following formula:*

Case $A \equiv (x = v)$ *where* x *is a variable and* $v \in \Psi(x)$.

$$\neg A \equiv (x \neq v)$$

Case *$A \equiv (x \neq v)$ where x is a variable and $v \in \Psi(x)$.*

$$\neg A \equiv (x = v)$$

Case *$A \equiv (x = y)$ where x and y are variables.*

$$\neg A \equiv (x \neq y)$$

Case *$A \equiv (x \neq y)$ where x and y are variables.*

$$\neg A \equiv (x = y)$$

Notation 3.1. *Rules serve as the conventional method for representing knowledge within an expert system and are incorporated into the preceding definition. Similar to Propositional Logic, we employ the notation of rules*

$$(A_1 \wedge ... \wedge A_r) \longrightarrow (B_1 \vee ... \vee B_s)$$

(where $A_1, ..., A_r, B_1, ...B_s$ are atomic formulae) to denote the formula:

$$\neg A_1 \vee ... \vee \neg A_r \vee B_1 \vee ... \vee B_s$$

We can now formally define the components of an expert system: Input, Knowledge-base and Output:

Definition 3.5. *We formally define the three components of an expert system as:*

- *Input. This is a finite set of positive atomic formulae, denoted as $\mathcal{F} \subset C$*
- *Knowledge-base. This is a set of formulae denoted as $\mathcal{K} \subset C$. The knowledge base comprises two types of formulae:*

  Integrity Constraints. *For each variable x_i that can only assume a finite set of potential values, i.e., $\Psi(x_i) = \{v_1 \ldots v_m\}$, the following formula represents the intrinsic integrity constraint:*

  $$(x_1 = v_1) \vee (x_1 = v_2) \vee \ldots \vee (x_1 = v_m)$$

  Rules. *These are formulae expressed via rules (provided by a human expert)*

- *Output. This is a finite set of atomic formulae.*

Figure 1 depicts the components of the expert system. As may be seen, integrity constraints are intuitively and implicitly derived from the inherent nature of variables and they are immediately obtained by the function Ψ of the conceptual framework (see Definition 3.1). For instance, a boolean variable, x, is associated with the integrity constraint $x = \text{False} \vee x = \text{True}$ because $\Psi(x) = \{\text{True}, \text{False}\}$, which signifies that x must assume one of two possible values: True or False. Similarly, if x denotes the current colour of a semaphore, we would have the integrity constraint $x = \text{red} \vee x = \text{orange} \vee x = \text{green}$, indicating that the semaphore's colour can be either red, orange, or green, which involves that $\Psi(x) = \{\text{red, orange, green}\}$. In Figure 1, the variable x_1, with $\Psi(x_1) = \{v_1, v_2, v_3\}$, is associated with the integrity constraint $A_1 \equiv (x_1 = v_1) \vee (x_1 = v_2) \vee (x_1 = v_3)$.

On the other hand, rules are formulas explicitly provided by a human expert. The acquisition of these rules is not a straightforward task. It necessitates a process that involves multiple interviews with experts, not only to elucidate these rules from them but also to validate the completeness and accuracy of the system. Although we provide a theorem that can be used to verify if the system is at least consistent (see Theorem 4.2), we do not delve into the process of elucidating these rules. Instead, our focus will be on the design of an inference engine based on an algebraic representation through polynomials, which allows for the automatic deduction of the system's output (as we will demonstrate in Section 4.2). In this way, the system's output is correct, as we will demonstrate, in the sense that it is deduced from the facts and rules.

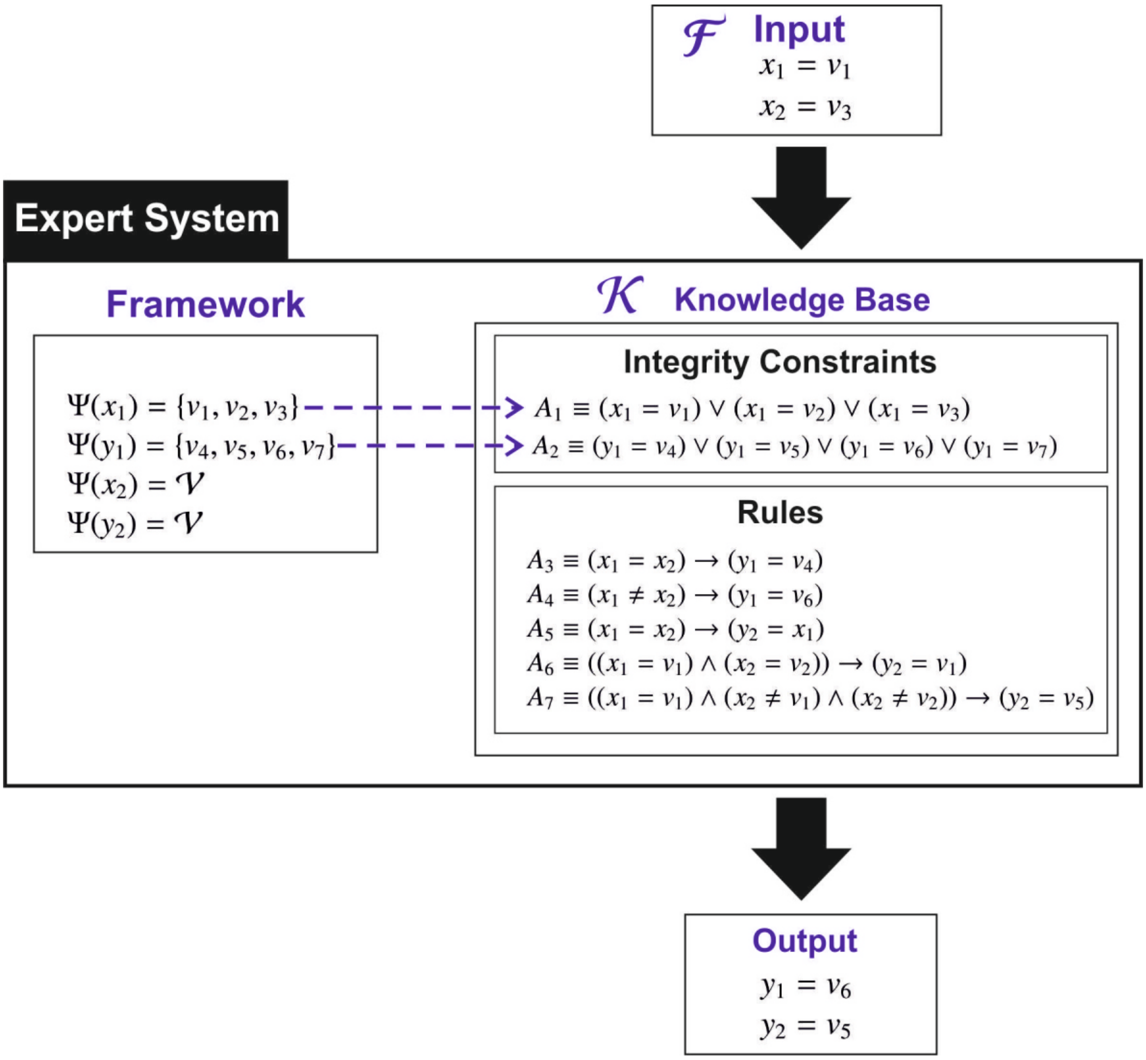

Figure 1. Components of the Expert Systems.

Next, we will establish the semantics of formulae:

Definition 3.6 (Holds)**.** *Let $A \in \mathcal{C}$ be a formula. Let $S \in \mathcal{S}$ be a state*
We say that the formula A holds in the state S if and only if:

Case $A \equiv x = v$ *where* $x \in \mathcal{X}$ *and* $v \in \Psi(x)$*.*

A holds in S $\Leftrightarrow S(x) = v$.

Case $A \equiv x \neq v$ *where* $x \in \mathcal{X}$ *and* $v \in \Psi(x)$.
A holds in S $\Leftrightarrow S(x) \neq v$.

Case $A \equiv x = y$ *where* $x, y \in \mathcal{X}$.
A holds in S $\Leftrightarrow S(x) = S(y)$.

Case $A \equiv x \neq y$ *where* $x, y \in \mathcal{X}$.
A holds in S $\Leftrightarrow S(x) \neq S(y)$.

Case $A \equiv B_1 \vee ... \vee B_r$ *where* $B_1, ..., B_r$ *are atomic formulae.*
A holds in S $\Leftrightarrow \exists i \in \{1, ..., r\}$ *such that* B_i *holds in* S.

Similar to Propositional logic, we need to establish the concepts of consistency and inference.

Definition 3.7 (Consistency)**.** *The set of formulae* $\{A_1, ..., A_n\}$ *is consistent if and only if* $\exists S \in \mathcal{S}$ *such that* $\forall i \in \{1, ..., n\}$ A_i *holds in* S.

Definition 3.8 (Derivable Formula)**.** *The formula* $B \in C$ *is derivable from the formulae* $A_1, ..., A_n$ *if and only if* $\forall S \in \mathcal{S}$ *in which all the formulae* A_1,...,A_n *hold, the formula* B *also holds in* S.

Analogous to Propositional Logic, the subsequent proposition is valid:

Proposition 3.1. *Let* $B = B_1 \vee ... \vee B_r$ *be a formula where* $B_1, ..., B_r$ *are atomic formulae. The formula* B *is derivable from* $A_1, ..., A_n$ *if and only if*

$$\forall i \in \{1, ..., r\} \text{ the set of formulae } \{A_1, ..., A_n, \neg B_i\} \text{ is not consistent.}$$

Let x be a variable which may take a finite set of values, represented as $\Psi(x) = \{v_1, ..., v_r\}$. The potential values of this variable can be expressed by the formula: $(x_i = v_1) \vee (x_i = v_2) \vee ... \vee (x_i = v_r)$. Such formulae are integral to the knowledge base of the expert system, $\mathcal{K}$.

Example 3.1. *To better illustrate the concepts we have discussed, let us consider a small example of the expert system depicted in Figure 1. We will define the set of variables as* $\mathcal{X} = \{x_1, x_2, y_1, y_2\}$, *and the potential values as* $\mathcal{V} = \{v_i | i \in \mathbb{N}\}$.

- *We will define the conceptual framework of the expert system (see Definition 3.1)*
  - *The variable* x_1 *can take any value of the set* $\{v_1, v_2, v_3\}$. *That is to say, we have that* $\Psi(x_1) = \{v_1, v_2, v_3\}$.
  - *The variable* x_2 *can take any value. That is to say, we have that* $\Psi(x_2) = \mathcal{V}$.
  - *The variable* y_1 *can take any value of the set* $\{v_4, v_5, v_6, v_7\}$. *That is to say, we have that* $\Psi(y_1) = \{v_4, v_5, v_6, v_7\}$.
  - *The variable* y_2 *can take any value. That is to say, we have that* $\Psi(y_2) = \mathcal{V}$.
- *Next, we will consider the knowledge-base in the system (see Definition 3.5).*

  Integrity Constrains. *Here, we will examine formulae derived from the potential set of values. We are dealing with only two variables,* x_1 *and* y_1, *each with a finite set of potential values. As a result, we can establish that:*

– *The variable* x_1 *may take the values* $\{v_1, v_2, v_3\}$.
  $A_1 \equiv (x_1 = v_1) \vee (x_1 = v_2) \vee (x_1 = v_3)$.
– *The variable* y_1 *may take the values* $\{v_4, v_5, v_6, v_7\}$.
  $A_2 \equiv (y_1 = v_4) \vee (y_1 = v_5) \vee (y_1 = v_6) \vee (y_1 = v_7)$.

Rules. *We will consider that the expert system in this example considers the following rules:*

$A_3 \equiv (x_1 = x_2) \rightarrow (y_1 = v_4)$
$A_4 \equiv (x_1 \neq x_2) \rightarrow (y_1 = v_6)$
$A_5 \equiv (x_1 = x_2) \rightarrow (y_2 = x_1)$
$A_6 \equiv ((x_1 = v_1) \wedge (x_2 = v_2)) \rightarrow (y_2 = v_1)$
$A_7 \equiv ((x_1 = v_1) \wedge (x_2 \neq v_1) \wedge (x_2 \neq v_2)) \rightarrow (y_2 = v_5)$

The knowledge-base of the expert system is $\mathcal{K} = \{A_1, ..., A_7\}$.

- *Let us consider that the input of our expert system is:*

$$\mathcal{F} = \{(x_1 = v_1), (x_2 = v_3)\}$$

- *Consider a potential state* S *(refer to Definition 3.2) of the system where every formula in* $\mathcal{K} \cup \mathcal{F}$ *is satisfied:*

$$S(x_1) = v_1; S(x_2) = v_3$$
$$S(y_1) = v_6; S(y_2) = v_5$$

It can be observed that all the formulas in $\mathcal{K} \cup \mathcal{F}$ *hold (see Definition 3.6). For instance, the formula (refer to Notation 3.1)*

$$A_7 \equiv ((x_1 = v_1) \wedge (x_2 \neq v_1) \wedge (x_2 \neq v_2)) \rightarrow (y_2 = v_5)$$

is equivalently written as:

$$A_7 \equiv (x_1 \neq v_1) \vee (x_2 = v_1) \vee (x_2 = v_2) \vee (y_2 = v_5)$$

Given that $S(y_2) = v_5$, *the formula* A_7 *holds in the state* S *as per Definition 3.6.*
Similarly, the remaining formulas in $\mathcal{K} \cup \mathcal{F}$ *hold.*

- *As can be easily deduced, the expert system outputs the formulae:*

$$y_1 = v_6$$
$$y_2 = v_5$$

We intuitively deduce them as follows:

– *Given* $x_1 \neq x_2$ *(since* $x_1 = v_1$ *and* $x_2 = v_3$*), by applying rule* A_4, *we conclude that* $y_1 = v_6$.
– *Given* $x_1 = v_1$, *and, since* $x_2 = v_3$, $x_2 \neq v_1$ *and* $x_2 \neq v_2$, *by applying rule* A_7, *we conclude that* $y_2 = v_5$.

Formally, we deduce $y_1 = v_6$ *and* $y_2 = v_5$ *because for every state where all formulae in* $\mathcal{K} \cup F$ *hold, the formulae* $y_1 = v_6$ *and* $y_2 = v_5$ *also hold (refer to Definition 3.8). In other words, for every state* S *such that* $S(x_1) = v_1$ *and* $S(x_2) = v_2$ *and the formulae in knowledge-base hold, it follows that* $S(y_1) = v_6$ *and* $S(y_2) = v_5$.

According to Proposition 3.1, the formula $y_1 = v_6$ is derived from input and knowledge-base as evidenced by the inconsistency of $\mathcal{K} \cup \mathcal{F} \cup \{y_1 \neq v_6\}$. Similarly, $y_2 = v_5$ is inferred due to the inconsistency of $\mathcal{K} \cup \mathcal{F} \cup \{y_2 \neq v_5\}$. In other words, there is no state S such that the formulae $\mathcal{K} \cup \mathcal{F}$ hold and $S(y_1) \neq v_6$; and there is no state S such that the formulae $\mathcal{K} \cup \mathcal{F}$ hold and $S(y_2) \neq v_5$ (refer to Definition 3.7).

4. The algebraic model

In this section, we will introduce an algebraic model that encapsulates the representation paradigm delineated in the preceding section. Utilizing this model, the following issues will be transposed into algebraic terms:

- The challenge of determining if there exists no possible state within a given knowledge base (refer to Theorem 4.2).
- The challenge of determining if a formula can be derived from the input and the knowledge-base (see Corollary 4.1).

4.1. Translation from the formulae into polynomials

Let us consider a set of variables $\mathcal{X} = \{x_1, ..., x_m\}$ and a set of formulae $\{A_1, ..., A_n\}$. Initially, we define a bijection, ϕ, that maps the possible values $\mathcal{V}$ to the field $\mathbb{Q}$.

$$\phi : \mathcal{V} \longrightarrow \mathbb{Q}$$

Each potential negative atomic formula (i.e., the formulae of the form $x_i \neq v_j$ or $x_i \neq x_j$) is associated with an auxiliary variable w_i. We assume that there are $w_1, ..., w_k$ auxiliary variables for representing the set of formulae $\{A_1, ..., A_n\}$.

Next, we define the polynomial ring:

$$\mathcal{A} = \mathbb{Q}\left[x_1, ..., x_m, w_1, ..., w_k, z\right]$$

Subsequently, we translate formulae into polynomials.

Definition 4.1 (Polynomial associated to a formula)**.** *For a formula $A \in C$, the polynomial $p_A \in \mathcal{A}$ associated to the formula A is defined as follows:*

Case *$A \equiv (x_i = v)$ where $x_i \in \mathcal{X}$ and $v \in \Psi(x_i)$.*
$p_A = x_i - \phi(v)$.

Case *$A \equiv (x_i = x_j)$ where $x_i, x_j \in \mathcal{X}$.*
$p_A = x_i - x_j$.

Case *$A \equiv (x_i \neq v)$ where $x_i \in \mathcal{X}$ and $v \in \Psi(x_i)$.*
$p_A = x_i + w - \phi(v)$ *where w_A is the variable associated to the formula $x_i \neq v$.*

Case *$A \equiv (x_i \neq x_j)$ where $x_i, x_j \in \mathcal{X}$.*
$p_A = x_i - x_j + w_A$ *where w_A is the variable associated to the formula $x_i \neq x_j$.*

Case *$A \equiv B_1 \vee ... \vee B_r$ where $B_1, ..., B_r$ are atomic formulae.*
$p_A = q_{B_1} \cdot ... \cdot q_{B_r}$

As we will explore later, each polynomial p_A represents an equation $p_A = 0$, which describes the semantics of the formula A. For instance, the formula $A \equiv (x_i = v)$ is associated with the polynomial $p_A = x_i - \phi(v)$, because $x_i - \phi(v) = 0$ if and only if $x_i = \phi(v)$. This corresponds to the semantics of the formula $A \equiv x_i = v$. It can be observed that a negative atomic formula involves the use of an auxiliary variable w_i. These variables w_i must assume values different from 0. Consequently, the formula $A \equiv (x_i = x_j)$ is associated with the polynomial $p_A = x_i - x_j + w_A$, because for $w_A \neq 0$, $p_A = 0$ if and only if $x_i \neq x_j$.

Nevertheless, it is possible to circumvent the use of this auxiliary variable when the negative atomic formula takes the form $x \neq v_1$ where x is a variable that can only assume a finite set of values $\Psi(x) = \{v_1, ..., v_r\}$. In this scenario, the formula is equivalent to (and therefore can be replaced by) the formula:

$$(x = v_2) \vee ... \vee (x = v_r)$$

This equivalence is particularly interesting when x assumes a small number of possible values. Specifically, in the case where x can only assume two possible values $\{v_1, v_2\}$ the negative atomic formula $x \neq v_1$ is equivalent to the atomic formula $x = v_2$.

This becomes especially noteworthy when considering a rule of the form:

$$(A_1 \wedge A_2 \wedge ... \wedge A_r) \rightarrow (B_1 \vee ... \vee B_s)$$

where $A_1, ...A_r, B_1, ..., B_s$ are atomic formulae. As mentioned above, this rule is expressed as the formula $\neg A_1 \vee ... \vee \neg A_r \vee B_1 \vee ... \vee B_s$ which corresponds to the polynomial:

$$p_{\neg A_1} \cdot ... \cdot p_{\neg A_r} \cdot p_{B_1} ... \cdot p_{B_s}$$

Therefore, in the scenario where $A_1, ..., A_r$ are negative atomic formulae and $B_1, ..., B_s$ are positive atomic formula, the rule does not necessitate any auxiliary variable to represent this polynomial. In Example 3.1, we have that the rule $A_4 \equiv (x_1 \neq x_2) \rightarrow (y_1 = v_6)$ is represented by $p_{A_4} = (x_1 - x_2)(y_1 - \phi(v_6))$. In the same way, since x_1 can only take a finite set of values (we have that $\Psi(x_1) = \{v_1, v_2, v_3\}$), we can state that the rule $A_7 \equiv ((x_1 = v_1) \wedge (x_2 \neq v_1) \wedge (x_2 \neq v_2)) \rightarrow (y_2 = v_5)$ is equivalent to the rule:

$$((x_1 \neq v_2) \wedge (x_1 \neq v_3) \wedge (x_2 \neq v_1) \wedge (x_2 \neq v_2)) \rightarrow (y_2 = v_5)$$

whose polynomial associated is:

$$(x_1 - \phi(v_2))(x_1 - \phi(v_3))(x_2 - \phi(v_1))(x_2 - \phi(v_2))(y_2 - \phi(v_5))$$

Example 4.1. *Consider the expert system described in Example 3.1. We define the bijection $\phi : \mathcal{V} \rightarrow C$ as follows:*

$$\phi(v_i) = i$$

Next, we computes the polynomials associated with the formulae in $\mathcal{K}$ (see Definition 4.1):

- $A_1 \equiv (x_1 = v_1) \vee (x_1 = v_2) \vee (x_1 = v_3)$
$$p_{A_1} = (x_1 - 1)(x_1 - 2)(x_1 - 3)$$
- $A_2 \equiv (y_1 = v_4) \vee (y_1 = v_5) \vee (y_3 = v_6) \vee (y_3 = v_7)$
$$p_{A_2} = (y_1 - 4)(y_1 - 5)(y_1 - 6)(y_1 - 7)$$
- $A_3 \equiv (x_1 = x_2) \to (y_1 = v_4)$
$$p_{A_3} = (x_1 - x_2 + w_1)(y_1 - 4)$$
- $A_4 \equiv (x_1 \neq x_2) \to (y_1 = v_6)$
$$p_{A_4} = (x_1 - x_2)(y_1 - 6)$$
- $A_5 \equiv (x_1 = x_2) \to (y_2 = x_1)$
$$p_{A_5} = (x_1 - x_2 + w_1)(y_2 - x_1)$$
- $A_6 \equiv ((x_1 = v_1) \wedge (x_2 = v_2)) \to (y_2 = v_1)$
$$p_{A_6} = (x_1 - 2)(x_1 - 3)(x_2 + w_2 - 2)(y_2 - 1)$$
- $A_7 \equiv ((x_1 = v_1) \wedge (x_2 \neq v_1) \wedge (x_2 \neq v_2)) \to (y_2 = v_5)$
$$p_{A_7} = (x_1 - 2)(x_1 - 3)(x_2 - 1)(x_2 - 2)(y_2 - 5)$$

where the auxiliary variables w_1 and w_2 are respectively associated to the formulae $(x_1 \neq x_2)$ and $(x_2 \neq v_2)$.

4.2. Deduction and consistence in the algebraic model

In this section we will in we will present some findings that recast the problem of verifying consistency and deduction in an expert system into algebraic terms (refer to Theorem 4.2 and Corollary 4.1). In the upcoming proposition, we will establish an initial relation between a formula and its corresponding polynomial.

Proposition 4.1. *Let $A(x_1, ..., x_m)$ be a formula. Let $S \in \mathcal{S}$ be a state. Let $p_A(x_1, ..., x_m, w_1, ..., w_k) \in \mathcal{A}$ be the polynomial associated to the formula A. We have that $A(x_1, ..., x_m)$ holds in S if and only if the following holds:*

$$\exists w_1^*, ..., w_k^* \in \mathbb{Q} - \{0\} \text{ such that } p_A(\phi(S(x_1)), ..., \phi(S(x_m)), w_1^*, ..., w_k^*) = 0$$

Proof. First, we will prove it when A is an atomic formula.

Case $A \equiv (x_i = v)$ where $v \in \mathcal{V}$.
We have that $p_A(x_i) = x_i - \phi(v)$.
$A(x_i)$ holds in $S \Leftrightarrow S(x_i) = v \Leftrightarrow \phi(S(x_i)) = \phi(v) \Leftrightarrow \phi(S(x_i)) - \phi(v) = 0 \Leftrightarrow p_A(\phi(S(x_i))) = 0$

Case $A \equiv (x_i \neq v)$ where $v \in \mathcal{V}$.
We have that $p_A(x_i, w_j) = x_i + w_j - \phi(v)$ where w_j is the auxiliary variable associated to A.
$A(x_i)$ holds in $S \Leftrightarrow S(x_i) \neq v \Leftrightarrow \phi(S(x_i)) \neq \phi(v) \Leftrightarrow$
$\Leftrightarrow \exists w_j^* \in \mathbb{Q} - \{0\}$ such that $\phi(S(x_i)) - \phi(v) + w_j^* = 0 \Leftrightarrow$
$\Leftrightarrow \exists w_j^* \in \mathbb{Q} - \{0\}$ such that $p_A(\phi(S(x_i)), w_j^*) = 0$

Case $A \equiv (x_i = x_j)$.

We have that $p_A(x_i, x_j) = x_i - x_j$.

$A(x_i, x_j)$ holds in $S \Leftrightarrow S(x_i) = S(x_j) \Leftrightarrow \phi(S(x_i)) = \phi(S(x_j)) \Leftrightarrow$

$\Leftrightarrow p_A(\phi(S(x_i)), \phi(S(x_j))) = 0$

Case $A \equiv (x_i \neq x_j)$.

We have that $p_A(x_i, x_j, w_s) = x_i - x_j + w_s$ where w_s is the auxiliary variable associated to A.

$A(x_i, x_j)$ holds in $S \Leftrightarrow S(x_i) \neq S(x_j) \Leftrightarrow \phi(S(x_i)) \neq \phi(S(x_j)) \Leftrightarrow$

$\Leftrightarrow \exists w_s^* \in \mathbb{Q} - \{0\}$ such that $\phi(S(x_i)) - \phi(S(x_j)) + w_s^* = 0 \Leftrightarrow$

$\Leftrightarrow \exists w_s^* \in \mathbb{Q} - \{0\}$ such that $p_A(\phi(S(x_i)), \phi(S(x_j)), w_s^*) = 0$

Now, we will prove for the case that A is a disjunction of atomic formulae. That is to say, $A \equiv B_1 \vee ... \vee B_r$ where $B_1, ..., B_r$ are atomic formulae. Therefore, we have that:
$p_A(x_1, ..., x_m, w_1, ..., w_k) = p_{B_1}(x_1, ..., x_m, w_1, ..., w_k) \cdot ... \cdot p_{B_r}(x_1, ..., x_m, w_1, ..., w_k)$.

In this case, we have that:
A holds in $S \Leftrightarrow \exists i \in \{1, ..., r\}$ such that B_i holds in $S \Leftrightarrow$
$\exists i \in \{1, ..., r\}\ \exists w_1^*, ..., x_k^* \in \mathbb{Q} - \{0\}\ p_{B_i}(\phi(S(x_1)), ..., \phi(S(x_m)), w_1^*, \dots w_k^*) = 0 \Leftrightarrow \exists w_1^*, ..., x_k^* \in \mathbb{Q} - \{0\}$
$p_{B_1}(\phi(S(x_1)), ..., \phi(S(x_m)), w_1^*, \dots w_k^*) \cdot ... \cdot p_{B_r}(\phi(S(x_1)), ..., \phi(S(x_m)), w_1^*, \dots w_k^*) = 0 \Leftrightarrow$
$p_A(\phi(S(x_1)), ..., \phi(S(x_1))) = 0$ □

Lemma 4.1. *Let* $A_1, ..., A_n \in C$ *be formulae.* $\{A_1, ..., A_r\}$ *is a consistent set of formulae if and only if* $\exists x_1^*, ..., x_m^*, w_1^*, ..., w_k^*, z^* \in \mathbb{Q}$ *such that*

$$\forall i \in \{1, ..., n\}\ p_{A_i}(x_1^*, ..., x_m^*, w_1^*, ..., w_k^*) = 0$$

$$1 + z^* \cdot w_1^* \cdot ... \cdot w_k^* = 0$$

Proof. $\Rightarrow$) Let $\{A_1, ..., A_n\}$ be a consistent set of formulae.
Let S be a state in which all the formulae $A_1, ..., A_n$ hold.
Let $x_i^* = \phi(S(x_i)) \in \mathbb{Q}$ where $i \in \{1, ..., m\}$.
According to Proposition 4.1, we have that $\exists w_1^*, ..., w_k^* \in \mathbb{Q} - \{0\}$ such that:

$$\forall i \in \{1, ..., r\}\ p_{A_i}(x_1^*, ..., x_m^*, w_1^*, ..., w_k^*) = 0$$

Since $\forall i \in \{1, ..., k\}\ w_i^* \neq 0$, we have that $\exists z^* \in \mathbb{Q}$ such that

$$1 + z^* \cdot w_1^* \cdot ... \cdot w_k^* = 0$$

$\Leftarrow$)Let $x_1^*, ..., x_m^*, w_1^*, ..., w_k^*, z^* \in \mathbb{Q}$ such that:

- $1 + z^* \cdot w_1^* \cdot ... \cdot w_k^* = 0$.
- $\forall i \in \{1, ..., n\}\ p_{A_i}(x_1^*, ..., x_m^*, w_1^*, ..., w_k^*) = 0$

Let S be the state such that $\forall i \in \{1, ..., m\}\ S(x_i) = \phi^{-1}(x_i^*)$.
Since $1 + z^* \cdot w_1^* \cdot ... \cdot w_k^* = 0$, we have that $\forall i \in \{1, ..., k\}\ w_i^* \neq 0$. That is to say $w_1^*, ..., w_k^* \in \mathbb{Q} - \{0\}$.
Since $\forall i \in \{1, ..., n\}\ p_{A_i}(x_1^*, ..., x_m^*, w_1^*, ..., w_k^*) = 0$, by Proposition 4.1, $\forall i \in \{1, ..., n\}\ A_i$ holds in S.
Consequently, we have that $\{A_1, ..., A_n\}$ is consistent. □

Theorem 4.2. *Let* $A_1, ..., A_n \in C$ *be formulae.*

$$\{A_1,...,A_n\} \textit{ is consistent} \Leftrightarrow 1 \notin \langle p_{A_1},...,p_{A_n},1+z\cdot w_1\cdot ...\cdot w_k\rangle$$

Proof.
Let $I = \langle p_{A_1},...,p_{A_n},1+z\cdot w_1\cdot ...\cdot w_k\rangle$

$\Rightarrow$) Suppose that $\{A_1,...,A_n\}$ is consistent. By Lemma 4.1, we have that $\exists x_1^*,\dots x_m^*, w_1^*,\dots w_k^*, z^* \in \mathbb{Q}$ such that

$$\forall i\ p_{A_i}(\phi(S(x_1))\dots\phi(S(x_m)), w_1^*\dots w_k^*) = 0$$

$$1+z^*\cdot w_1^*\cdot w_k^* = 0$$

We will establish this proof by employing a reductio ad absurdum argument. Let's assume that $1 \in I$. Under this assumption, we would have

$$1 = \alpha_1 p_{A_1} + ... + \alpha_n p_{A_n} + \alpha_{n+1}(1+z\cdot w_1\cdot ...\cdot w_k)$$

Therefore, we have that

$$1 = 1(\phi(S(x_1))\dots\phi(S(x_m)), w_1^*\dots w_k^*) =$$

$$= \alpha_1 p_{A_1} + ... + \alpha_n p_{A_n} + \alpha_{n+1}(1+z\cdot w_1\cdot ...\cdot w_k)(\phi(S(x_1))\dots\phi(S(x_m)), w_1^*\dots w_k^*) = 0$$

This leads to a contradiction. Therefore, we must conclude that $1 \notin I$.

$\Leftarrow$) We will consider that $\{A_1,...,A_n\}$ is inconsistent. We will consider that all the polynomials $p_{A_1}\dots p_{A_n}, 1+z\cdot w_1\cdot ...\cdot w_k$ lie in $\mathbb{C}[x_1\dots x_n, w_1\dots w_k, z] \subset \mathbb{Q}[x_1\dots x_n, w_1\dots w_k, z]$. Since $\{A_1,...,A_n\}$ is inconsistent, we have that $\nexists x_1^*,\dots x_m^*, w_1^*,\dots w_k^*, z^* \in \mathbb{Q}$ such that

$$\forall i\ p_{A_i}(\phi(S(x_1))\dots\phi(S(x_m)), w_1^*\dots w_k^*) = 0$$

$$1+z^*\cdot w_1^*\cdot w_k^* = 0$$

Since p_{A_i} is the product of simple factors, it is immediate to state that: $\nexists x_1^*,\dots x_m^*, w_1^*,\dots w_k^*, z^* \in \mathbb{C}$ such that

$$\forall i\ p_{A_i}(\phi(S(x_1))\dots\phi(S(x_m)), w_1^*\dots w_k^*) = 0$$

$$1+z^*\cdot w_1^*\cdot w_k^* = 0$$

Consequently, we ascertain that $V(I) = \emptyset$ and, by applying the weak Hilbert's Nullstellensatz, we infer that $I = \langle 1\rangle$. This leads us to the conclusion that $1 \in I$. □

Corollary 4.1. *A formula B is derivable from* $\{A_1\dots A_n\}$ *if and only if*

$$1 \in \langle p_{\neg B}, p_{A_1},\dots p_{A_n}, 1+z\cdot w_1\cdot ...\cdot w_k\rangle$$

Proof. This is an immediate consequence of Proposition 3.1 and Theorem 4.2. □

4.3. Our algebraic model for Expert Systems

In Figure 2 we illustrate how we can implement the expert systems developed in Section 3 by means of the mathematical results obtained previously.

Formulae in the input and the knowledge base are represented by means of polynomials according to Definition 4.1, resulting in the ideal F and K:

Ideal F. This is the ideal generated by the polynomials representing the formulae in the input $\mathcal{F}$.

Ideal K. This is the ideal generated by the polynomials representing the formulae in the knowledge-base $\mathcal{K}$.

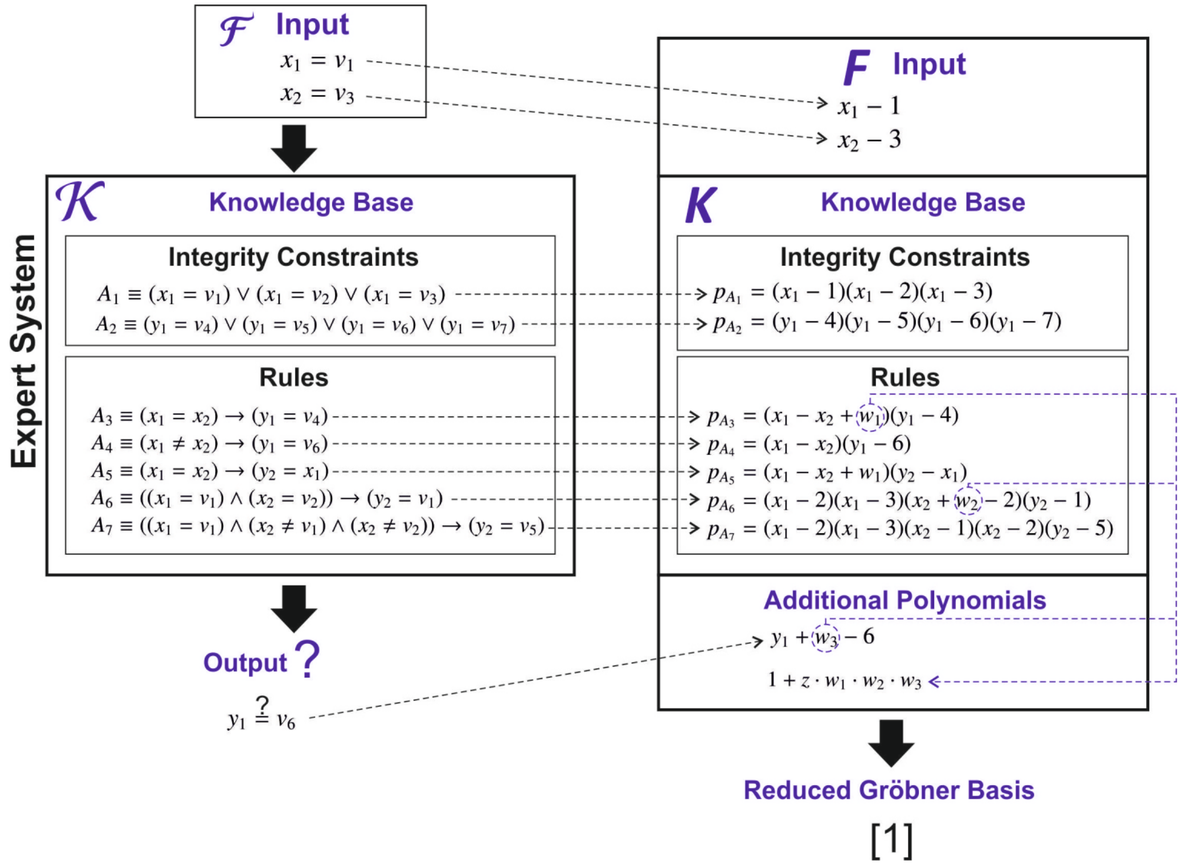

Figure 2. Our algebraic approach for implementing Expert Systems.

In accordance with Corollary 4.1, the expert system infers the formula B if and only if

$$1 \in K + F + \langle p_{\neg B}, 1 + z \cdot w_1 \cdot ... \cdot w_k \rangle$$

where $w_1 \dots w_k$ are the auxiliary variables utilized to represent the formulae in $\mathcal{K}$, $\mathcal{F}$ and the formula $\neg B$.

Consequently, we need to consider two additional polynomials (see Figure 2):

- The polynomial $1 + z \cdot w_1 \cdot ... \cdot w_k$ associated to the auxiliary variables w's needed to represent negative atomic formulae.

- The polynomial p_B associated to the atomic formula B we wish to determine if the system outputs.

According to this corollary, we can determine if the system outputs B by calculating the reduced Gröbner basis of the ideal generated by all previous polynomials (that is to say, the reduced Gröbner basis of the ideal $K + F + \langle p_{\neg B}, 1 + z \cdot w_1 \cdot ... \cdot w_k \rangle$) and examining whether it equals [1]. If it does, expert system outputs B.

Example 4.2. *Let us consider the expert system described in Example 3.1 and Example 4.1, and depicted in Figure 2. By applying Theorem 4.2, we will verify that the output of the expert system is* $(y_1 = v_6)$ *and* $y_2 = v_5$ *when the input of the expert system is* $\mathcal{F} = \{(x_1 = v_1), (x_2 = v_3)\}$.

We will stop, as an example, at the polynomials associated with some formulae (see Definition 4.1) in the input, $\mathcal{F}$*, and the knowledge-base,* $\mathcal{K}$*:*

- *The atomic formula in the input* $x_1 = v_1$ *corresponds to the polynomial:*

$$x_1 - 1$$

- *The integrity constraint* $A_1 \equiv (x_1 = v_1) \vee (x_1 = v_2) \vee (x_1 = v_3)$ *corresponds to the polynomial:*

$$p_{A_1} = (x_1 - 1)(x_1 - 2)(x_1 - 3)$$

- *The rule* $A_4 \equiv (x_1 \neq x_2) \to (y_1 = v_6)$ *corresponds to the polynomial:*

$$p_{A_4} = (x_1 - x_2)(y_1 - 6)$$

  Note that the formula A_4 *can be also written (see Notation 3.1) as* $A_4 \equiv (x_1 = x_2) \vee (y_1 = v_6)$
- *The formula* $A_3 \equiv (x_1 = x_2) \to (y_1 = v_4)$ *corresponds to the polynomial:*

$$p_{A_3} = (x_1 - x_2 + w_1)(y_1 - 4)$$

  Note that the formula A_3 *can be also written (see Notation 3.1) as* $A_3 \equiv (x_1 \neq x_2) \to (y_1 = v_4)$ *and we need an auxiliary variable,* w_1*, for the atomic formula* $x_1 \neq x_2$.

In this way, we have that:

- *The ideal associated to the input is*

$$F = \langle x_1 - 1, x_2 - 3 \rangle$$

- *The ideal* K *associated with the knowledge-base of the expert system is:*

$$K = \langle p_{A_1}, p_{A_2}, p_{A_3}, p_{A_4}, p_{A_5}, p_{A_6}, p_{A_7} \rangle$$

Besides, we need two additional polynomials:

- *The polynomial associated to* $\neg B$*. If the output formula* B *is* $y_1 = v_6$*, we have that* $p_{\neg B}$ *is:*

$$y_1 + w_3 - 6$$

  where w_3 *is an auxiliary variable used for the negative atomic formula* $\neg B \equiv y_1 \neq v_6$
- *The polynomial associated to all the auxiliary variables w's used:*

$$1 + z \cdot w_1 \cdot w_2 \cdot w_3$$

To verify whether the system can infer $y_1 = v_6$*, we need to check if the reduced Gröbner basis of the ideal generated by the previous polynomials equals* [1]*:*

$$K + F + \langle y_1 + w_3 - 6, 1 + z \cdot w_1 \cdot w_2 \cdot w_3 \rangle$$

5. Intuition of our approach

In this section, we strive to clarify the underlying logic of our approach. The inference engine is fundamentally based on an algebraic approach. Both rules and facts are interconnected through algebraic equations that a state compatible with formulae must satisfy. Each polynomial p used to infer the output of the system is associated with the algebraic equation $p = 0$. Finding a state S compatible with all the formulae is equivalent to solve a this set of algebraic equations. In Figure 3 we illustrate the set of equations generated by the expert system in Example 3.1, which are related to the polynomials used in Example 4.2. For example:

- The formula $A_1 \equiv (x_1 = v_1) \vee (x_1 = v_2) \vee (x_1 = v_3)$ corresponds to the polynomial:

$$p_{A_1} = (x_1 - 1)(x_1 - 2)(x_1 - 3)$$

  Note that $p_{A_1} = (x_1 - 1)(x_1 - 2)(x_1 - 3) = 0$ if and only if either $x_1 = 1$ or $x_1 = 2$ or $x_1 = 3$, which aligns with the semantics of the formula A_1.
- The formula $A_4 \equiv (x_1 \neq x_2) \to (y_1 = v_6)$ corresponds to the polynomial:

$$p_{A_4} = (x_1 - x_2)(y_1 - 6)$$

  Note that $p_{A_4} = (x_1 - x_2)(y_1 - 6) = 0$ if and only if either $x_1 = x_2$ or $y_1 = 6$. In other words, if $(x_1 \neq x_2)$ then y_1 must be 6, which aligns with the semantics of the formula $A_4 \equiv (x_1 \neq x_2) \to (y_1 = v_6)$.
- The formula $A_3 \equiv (x_1 = x_2) \to (y_1 = v_4)$ corresponds to the polynomial:

$$p_{A_3} = (x_1 - x_2 + w_1)(y_1 - 4)$$

  where w_1 must be a value different from 0.
  Given that $w_1 \neq 0$, note that $p_{A_3} = (x_1 - x_2 + w_1)(y_1 - 4) = 0$ if and only if either $x_1 \neq x_2$ (since $w_1 \neq 0$) or $y_1 = 4$. In other words, if $x_1 = x_2$ then y_1 must be 4, which aligns with the semantics of the formula $A_3 \equiv (x_1 = x_2) \to (y_1 = v_4)$.

In Example 4.2, we have the input $\mathcal{F} = \{x_1 = v_1, x_2 = v_3\}$, represented by the polynomials:

- The fact $x_1 = v_1$ corresponds to the polynomial:

$$x_1 - 1$$

  Note that $x_1 - 1 = 0$ if and only if $x_1 = 1$, which aligns with the semantics of the fact $x_1 = v_1$.
- The fact $x_2 = v_3$ corresponds to the polynomial:

$$x_2 - 3$$

  Note that $x_2 - 3 = 0$ if and only if $x_2 = 3$, which aligns with the semantics of the fact $x_2 = v_3$.

To verify if the system outputs $y_1 = v_6$, we need to represent the formula $y_1 \neq v_6$. This is represented by the polynomial:

$$y_1 + w_3 - 6$$

where $w_3 \neq 0$. Note that $y_1 + w_3 - 6 = 0$ if and only if $y_1 \neq 6$ (since $w_3 \neq 0$), which aligns with the semantics of the formula $y_1 = v_6$.

Additionally, we have another polynomial associated with the variables w_i:

$$1 + z \cdot w_1 \cdot w_2 \cdot w_3$$

Note that $1 + z \cdot w_1 \cdot w_2 \cdot w_3 = 0$ if and only if $w_1 \neq 0$ and $w_2 \neq 0$ and $w_3 \neq 0$. Consequently, this equation associated with this polynomial ensures that the variables w must be different from 0.

In this way, we have the following set of equations:

- The set of equations related to $\mathcal{K}$:

$$\begin{gathered}(x_1 - 1)(x_1 - 2)(x_1 - 3) = 0\\(y_1 - 4)(y_1 - 5)(y_1 - 6)(y_1 - 7) = 0\\(x_1 - x_2 + w_1)(y_1 - 4) = 0\\(x_1 - x_2)(y_1 - 6) = 0\\(x_1 - x_2 + w_1)(y_2 - x_1) = 0\\(x_1 - 2)(x_1 - 3)(x_2 + w_2 - 2)(y_2 - 1) = 0\\(x_1 - 2)(x_1 - 3)(x_2 - 1)(x_2 - 2)(y_2 - 5) = 0\end{gathered}$$

- The set of equations related to $\mathcal{F}$:

$$\begin{gathered}x_1 - 1 = 0\\x_2 - 3 = 0\end{gathered}$$

- The equation related to the polynomial associated with the set of variables w's:

$$1 + z \cdot w_1 \cdot w_2 \cdot w_3 = 0$$

- The equation related to the output:

$$y_1 + w_3 - 6 = 0$$

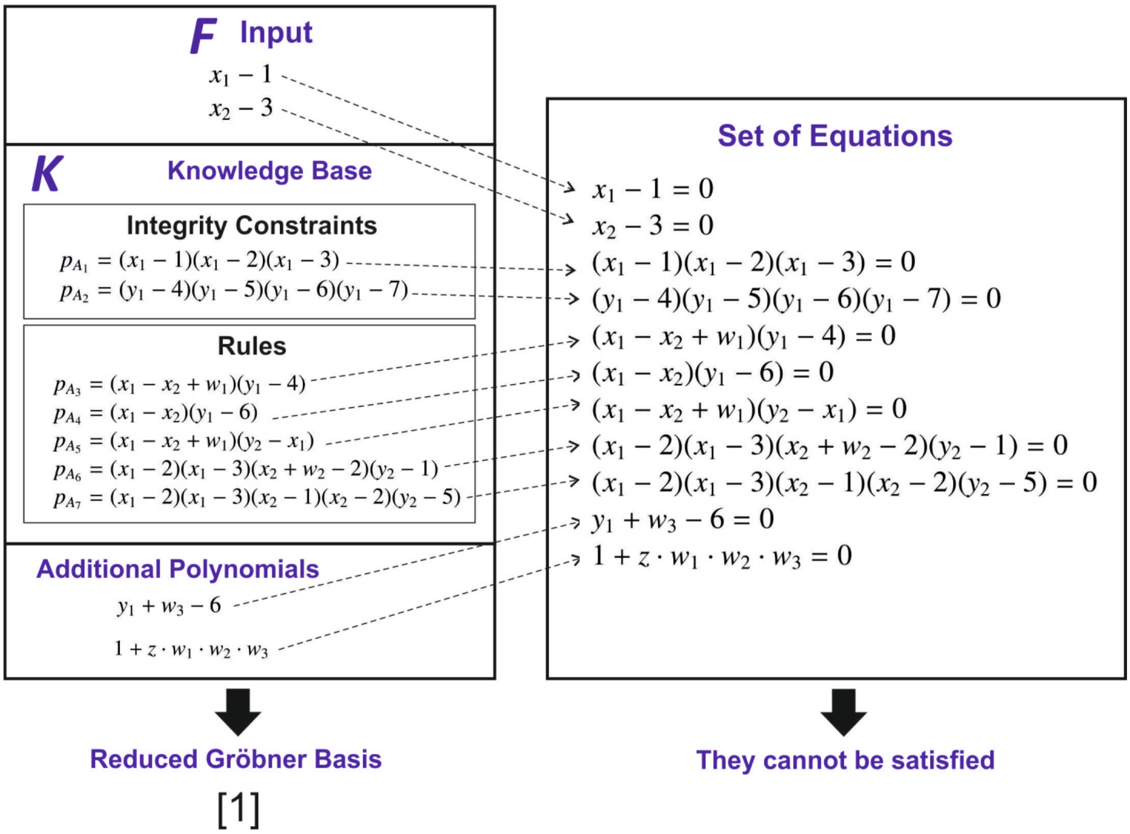

Figure 3. Our algebraic approach for implementing Expert Systems.

Through algebraic methods (by verifying that the reduced Gröbner basis is [1]), we conclude that it is unfeasible to find values for the variables x_1, x_2, y_1, y_2 that would satisfy all the preceding equations. Thus, if we discovered values for the variables x_1, x_2, y_1, y_2 that satisfied the equations of $\mathcal{K} \cup \mathcal{F} \cup \{1 + z \cdot w_1 \cdot w_2 \cdot w_3 = 0\}$ (in other words, the formulae of the knowledge base and the input hold), then the equation $y_1 + w_3 - 6 = 0$ would not be satisfied (in other words, the formula associated to the output does not hold). Considering that $w_3 \neq 0$ (as the equation $1 + z \cdot w_1 \cdot w_2 \cdot w_3 = 0$ is satisfied), y_1 must be 6 for the equation $y_1 + w_3 - 6 = 0$ to not be satisfied. In other words, we infer that y_1 must be equal to v_6. In summary, our analysis of the previous equations leads us to deduce that $y_1 = v_6$ is a consequence of the input and the knowledge base.

6. Example: Trains

In this section, we will explore the potential of designing an interlocking problem for a railway system using our algebraic approach. An interlocking system is a safety-critical mechanism engineered to prevent train collisions. Various approaches have been proposed by researchers have to address the problem of determining if two trains may collide [23–29]. In this section we will easily design an interlocking system by means of the paradigm described here: An interlocking system as an expert system whose variables assume an infinite set of potential values. This approach allows for a straightforward design process.

Given a railway station, our goal is to design an expert system that can determine whether a given situation poses a danger. We will illustrate the concepts of our paper using the railway station depicted in Figure 4. However, it is important to note that these principles can be generalized to any railway station.

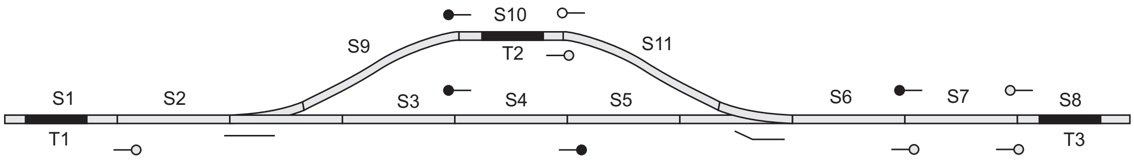

Figure 4. Dangerous Situation in a Railway station.

6.1. Railway Interlocking Systems

We will be examining the railway station depicted in Figure 4. As can be observed, it comprises:

- 11 sections,
- Two turnouts: one turnout connecting S2 to sections S3 or S4 (since the switch is on direct track position, trains move from S2 to S3), and another connecting S6 to sections S5 or S11 (in this case, since the switch is on diverted track position, trains move from S6 to S11)
- 10 semaphores depicted by cycles (black representing red colour, white representing green colour)
- Three trains, T1, T2 and T3, placed respectively in sections S1, S10 and S8.

As may be seen, the situation depicted in Figure 4 is dangerous: the trains situated in sections S10 and S8 could collide in section S7: Train in S10 moves from S10 to S11, then to S6 and finally from S6 to S7; train in S8 moves from S8 to S7.

6.2. The Expert System

6.2.1. Conceptual framework

We will consider a railway station that has N sections, denoted as $S_1 \dots S_N$, and trains in the station are identified by a natural number. We will consider that $\mathcal{V} = \mathbb{N}$.

The set of variables, $\mathcal{X}$ consists of two types:

A variable $e_{i,j}$ for each two pair of connected sections S_i and S_j. For any two sections S_i and S_j that are connected at the edge, we will define the variable $e_{i,j} \in \{0, 1\}$, i.e. $\Psi(e_{i,j}\{0, 1\}$. The variable $e_{i,j} = 1$ indicates that it is possible for a train to pass from section S_i to section S_j and $e_{i,j} = 0$ indicates that it is not possible to pass from section s_i to section s_j.

In the railway station depicted in Figure 4, we would have the following variables:
$e_{1,2}, e_{2,9}, e_{9,10}, e_{10,11}, e_{11,6}, e_{2,3}, e_{3,4}, e_{4,5}, e_{5,6}, e_{6,7}, e_{7,8}$,
$e_{2,1}, e_{9,2}, e_{10,9}, e_{11,10}, e_{6,11}, e_{3,2}, e_{4,3}, e_{5,4}, e_{6,5}, e_{7,6}, e_{8,7}$

A variable x_i for each section S_i in the railway station . We define $\Psi(x_i) = \mathcal{V} = \mathbb{N}$. The value $x_i = t_j$ signifies that train $t_j \in \mathbb{N}$ can reach the section S_i. In no trains can reach section S_i, then we set $x_i = 0$. If a situation is dangerous, there would be two different trains t_j and t_k that could reach the same section S_j. As a result, we would have the formulae $x_i = t_j$ and $x_i = t_k \neq t_j$, which are inconsistent formulae. In the railway station depicted in Figure 4, we would have the following variables:

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}$$

6.2.2. Knowledge-Base

We will consider the following formulae in the knowledge-base, $\mathcal{K}$:

Integrity Constraints. These are formulae related to the possible values, $\{0, 1\}$, that variables $e_{i,j}$ may assume. Specifically, for every pair of sections i and j we have that $(e_{i,j} = 0) \vee (e_{i,j} = 1)$. Consequently, we have that: $e_{1,2} = 0 \vee e_{1,2} = 1$
$e_{2,9} = 0 \vee e_{2,9} = 1$
$e_{9,10} = 0 \vee e_{9,10} = 1$
$e_{10,11} = 0 \vee e_{10,11} = 1$
$e_{11,6} = 0 \vee e_{11,6} = 1$
$e_{2,3} = 0 \vee e_{2,3} = 1$
$e_{3,4} = 0 \vee e_{3,4} = 1$
$e_{4,5} = 0 \vee e_{4,5} = 1$
$e_{5,6} = 0 \vee e_{5,6} = 1$
$e_{6,7} = 0 \vee e_{6,7} = 1$
$e_{7,8} = 0 \vee e_{7,8} = 1$
$e_{2,1} = 0 \vee e_{2,1} = 1$
$e_{9,2} = 0 \vee e_{9,2} = 1$
$e_{10,9} = 0 \vee e_{10,9} = 1$
$e_{11,10} = 0 \vee e_{11,10} = 1$
$e_{6,11} = 0 \vee e_{6,11} = 1$
$e_{3,2} = 0 \vee e_{3,2} = 1$
$e_{4,3} = 0 \vee e_{4,3} = 1$
$e_{5,4} = 0 \vee e_{5,4} = 1$
$e_{6,5} = 0 \vee e_{6,5} = 1$
$e_{7,6} = 0 \vee e_{7,6} = 1$
$e_{8,7} = 0 \vee e_{8,7} = 1$

Rules. These are formulae related to the possible movements of trains. Given two sections S_i and S_j which may be connected, we will consider the following rule*:
$(e_{i,j} = 1) \wedge (x_i \neq 0) \rightarrow (x_j = x_i)$

Consequently, we have the following rules:
$(e_{1,2} = 1) \wedge (x_1 \neq 0) \rightarrow (x_2 = x_1)$
$(e_{2,9} = 1) \wedge (x_2 \neq 0) \rightarrow (x_9 = x_2)$
$(e_{9,10} = 1) \wedge (x_9 \neq 0) \rightarrow (x_{10} = x_9)$
$(e_{10,11} = 1) \wedge (x_{10} \neq 0) \rightarrow (x_{11} = x_{10})$

*This rule implies that if the train T_k may reach section S_i (i.e., $x_i = k \neq 0$), and it is possible to pass from section S_i to section S_j, then the same train T_k can reach section S_j (i.e., $x_j = k = x_i$).

$(e_{11,6} = 1) \wedge (x_{11} \neq 0) \rightarrow (x_6 = x_{11})$
$(e_{2,3} = 1) \wedge (x_2 \neq 0) \rightarrow (x_3 = x_2)$
$(e_{3,4} = 1) \wedge (x_3 \neq 0) \rightarrow (x_4 = x_3)$
$(e_{4,5} = 1) \wedge (x_4 \neq 0) \rightarrow (x_5 = x_4)$
$(e_{5,6} = 1) \wedge (x_5 \neq 0) \rightarrow (x_6 = x_5)$
$(e_{6,7} = 1) \wedge (x_6 \neq 0) \rightarrow (x_7 = x_6)$
$(e_{7,8} = 1) \wedge (x_7 \neq 0) \rightarrow (x_8 = x_7)$
$(e_{2,1} = 1) \wedge (x_2 \neq 0) \rightarrow (x_1 = x_2)$
$(e_{9,2} = 1) \wedge (x_9 \neq 0) \rightarrow (x_2 = x_9)$
$(e_{10,9} = 1) \wedge (x_{10} \neq 0) \rightarrow (x_9 = x_{10})$
$(e_{11,10} = 1) \wedge (x_{11} \neq 0) \rightarrow (x_{10} = x_{11})$
$(e_{6,11} = 1) \wedge (x_6 \neq 0) \rightarrow (x_1 1 = x_6)$
$(e_{3,2} = 1) \wedge (x_3 \neq 0) \rightarrow (x_2 = x_3)$
$(e_{4,3} = 1) \wedge (x_4 \neq 0) \rightarrow (x_3 = x_4)$
$(e_{5,4} = 1) \wedge (x_5 \neq 0) \rightarrow (x_4 = x_5)$
$(e_{6,5} = 1) \wedge (x_6 \neq 0) \rightarrow (x_5 = x_6)$
$(e_{7,6} = 1) \wedge (x_7 \neq 0) \rightarrow (x_6 = x_7)$
$(e_{8,7} = 1) \wedge (x_8 \neq 0) \rightarrow (x_7 = x_8)$

6.2.3. Input

The input is intrinsically linked to the status of the turnouts and semaphores within the railway station, as well as the positioning of the trains. As illustrated in Figure 4, the input is as follows:

Variables e_{ij}. For each pair of connected sections S_i, S_j, we have either an atomic formula ($e_{i,j} = 1$) or ($e_{i,j} = 0$) representing whether it is possible to transition from section S_i to section S_j. The colour of the semaphores and the position of turnout switches. For example, $e_{1,2} = 1$ in Figure 4 because the semaphore between section S1 to S2 is green. The turnout switch connecting sections S2, S3 and S9 being on direct results in $e_{2,3} = 1$ and $e_{2,9} = 0$: We have the following:
$e_{1,2} = 1; e_{9,10} = 1; e_{10,11} = 1; e_{11,6} = 1; e_{2,3} = 1; e_{3,4} = 1; e_{5,6} = 1; e_{6,7} = 1; e_{7,8} = 1;$
$e_{2,1} = 1; e_{11,10} = 1; e_{6,11} = 1; e_{3,2} = 1; e_{5,4} = 1; e_{8,7} = 1;$
$e_{2,9} = 0; e_{4,5} = 0; e_{9,2} = 0; e_{10,9} = 0; e_{4,3} = 0; e_{6,5} = 0; e_{7,6} = 0;$

Variables x_i. For each train T_j located in section S_i, we consider the positive atomic formula $x_i = j$. Given that there are three trains positioned on S1, S10 and S8 in Figure 4, we have that:
$x_1 = 1$
$x_{10} = 2$
$x_8 = 3$

6.2.4. Output

The primary objective of the system is to ensure the safety of the railway. If the railway is deemed unsafe, it two trains, represented as T_j and T_k, could potentially arrive at the same section S_i. In such a case, we would have that $x_i = t_j$ and $x_i = t_k$, and since $t_j \neq t_k$, it would lead to a contradiction in the set of formulae derived from the input and the knowledge base (as we can infer that $x_i = k \neq j = x_i$).

Therefore, the output mainly involves checking the consistency of the set of formulae generated by the input and the knowledge base. A consistent set indicates a safe situation, while an inconsistent set signifies danger.

6.3. *Implementation in* CoCoA

According to Section 4, we have that the polynomial ring is:

$$\begin{aligned}\mathcal{A} &= \mathbb{Q}[e_{1,2}, e_{2,9}, e_{9,10}, e_{10,11}, e_{11,6}, e_{2,3}, e_{3,4}, e_{4,5}, e_{5,6}, e_{6,7}, e_{7,8}, e_{2,1}, e_{9,2}, e_{10,9}, e_{11,10}, e_{6,11}, e_{3,2}, e_{4,3}, e_{5,4},\\ &\quad e_{6,5}, e_{7,6}, e_{8,7} x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}]\end{aligned}$$

In CoCoA syntax, we would have:

```
use QQ[e1_2, e2_9, e9_10, e10_11, e11_6, e2_3, e3_4, e4_5, e5_6, e6_7, e7_8,
e2_1, e9_2, e10_9, e11_10, e6_11, e3_2, e4_3, e5_4, e6_5, e7_6, e8_7, x[1..11]];
```

We will convert the rules and the integrity constraints of the system's knowledge base, denoted as $\mathcal{K}$, into polynomials. As per 4.1:

- The integrity constraint $(e_{i,j} = 0) \vee (e_{i,j} = 1)$ is represented by the polynomial

$$e_{i,j} \cdot (e_{ij} - 1)$$

- The rule $(e_{i,j} = 1) \wedge (x_i \neq 0) \rightarrow (x_j = x_i)$ is represented by the polynomial:

$$e_{i,j} \cdot x_i \cdot (x_j - x_i)$$

We define the ideal K as the ideal generated by the polynomials that represent these formulae in $\mathcal{K}$.

```
K:=Ideal(
e1_2 * (e1_2 - 1), e2_9 * (e2_9 - 1),
e9_10 * (e9_10 - 1), e10_11 * (e10_11 - 1),
e11_6 * (e11_6 - 1), e2_3 * (e2_3 - 1),
e3_4 * (e3_4 - 1), e4_5 * (e4_5 - 1),
e5_6 * (e5_6 - 1), e6_7 * (e6_7 - 1),
e7_8 * (e7_8 - 1), e2_1 * (e2_1 - 1),
e9_2 * (e9_2 - 1), e10_9 * (e10_9 - 1),
e11_10 * (e11_10 - 1), e6_11 * (e6_11 - 1),
e3_2 * (e3_2 - 1), e4_3 * (e4_3 - 1),
e5_4 * (e5_4 - 1), e6_5 * (e6_5 - 1),
e7_6 * (e7_6 - 1), e8_7 * (e8_7- 1),
e1_2 * x[1] * (x[2] - x[1]), e2_9 * x[2] * (x[9] - x[2]),
e9_10 * x[9] * (x[10] - x[9]), e10_11 * x[10] * (x[11] - x[10]),
e11_6 * x[11] * (x[6] - x[11]), e2_3 * x[2] * (x[3] - x[2]),
e3_4 * x[3] * (x[4] - x[3]), e4_5 * x[4] * (x[5] - x[4]),
e5_6 * x[5] * (x[6] - x[5]), e6_7 * x[6] * (x[7] - x[6]),
e7_8 * x[7] * (x[8] - x[7]), e2_1 * x[2] * (x[1] - x[2]),
```

```
e9_2 * x[9] * (x[2] - x[9]), e10_9 * x[10] * (x[9] - x[10]),
e11_10 * x[11] * (x[10] - x[11]), e6_11 * x[6] * (x[11] - x[6]),
e3_2 * x[3] * (x[2] - x[3]), e4_3 * x[4] * (x[3] - x[4]),
e5_4 * x[5] * (x[4] - x[5]), e6_5 * x[6] * (x[5] - x[6]),
e7_6 * x[7] * (x[6] - x[7]), e8_7 * x[8] * (x[7] - x[8])
);
```

For the situation depicted in Figure 4, the input is as follows:

```
F:=Ideal(e1_2 - 1, e9_10 - 1, e10_11 - 1, e11_6 - 1, e2_3 - 1, e3_4 - 1,
 e5_6 - 1, e6_7 - 1, e7_8 - 1, e2_1 - 1, e11_10 - 1, e6_11 - 1, e3_2 - 1,
 e5_4 - 1, e8_7 - 1, e2_9, e4_5, e9_2, e10_9, e4_3, e6_5, e7_6,
 x[1]-1, x[10]-2, x[8]-3);
```

According to Theorem 4.2, we can determine if the set of formulae $\mathcal{K}\cup\mathcal{F}$ is inconsistent (indicating a dangerous situation), by checking if the reduced Gröbner basis of the ideal $F + K$ is [1]. In CoCoA syntax, this is represented as:

```
ReducedGBasis(F+K)=[1];
```

Since *CoCoA* outputs true. Consequently, the situation is dangerous.

7. Conclusions

In this paper, we introduce an innovative algebraic methodology for the development of expert systems that can accommodate attributes capable of assuming a value from an infinite set. Prior algebraic approaches were reliant on representation formalisms based on either Propositional Logic or the Concept-Attribute-Value paradigm. Both of these formalisms necessitate that attributes assume a finite set of values. Despite the differences between these two approaches, they exhibit a certain degree of equivalence: Any expert system that can be algebraically implemented using one model can also be implemented using the other.

However, in scenarios where an attribute can assume a value from an infinite (or very large finite) set, the expert system cannot be represented using propositional logic. Until now, no method had been identified to implement such a system on a Computer Algebra System. This paper breaks new ground by presenting a model that not only addresses this gap but also provides a fresh perspective on previous results. In fact, these prior results can be viewed as specific instances within the broader framework of our proposed solution.

Our methodology can be employed in expert systems across various applications where uncertainty is not a factor. It is particularly suited for decision trees that culminate in a finite number of outputs, as opposed to intermediate steps and results with variable levels of certainty [12] (in fields such as medical diagnostics, quality improvement and business decision-making, among others). Indeed, our approach does present some limitations when variables assume a value from an infinite set since we have only considered relations between variables with equality ($x_i = x_j$) and inequality ($x_i \neq x_j$). The representation of order relations such as $x_i > x_j$ is currently not supported. Despite these limitations, our methodology offers a framework that can be applied in situations where systems do not utilize uncertainty. In this paper, we have demonstrated a practical application of our approach in determining interlocking problems.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

Professor José Luis Galán-García is a Guest Editor for AIMS Mathematics and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

References

1. A. Hernando, E. Roanes-Lozano, L. M. Laita, A polynomial model for logics with a prime power number of truth values, *J. Autom. Reasoning*, **46** (2011), 205–221. https://doi.org/10.1007/s10817-010-9191-0
2. J. Hsiang, Refutational Theorem Proving using Term-Rewriting Systems, *Artif. Intell.*, **25** (1985), 255–300.
3. 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), **2** (1985), 1146–1153.
4. J. Chazarain, A. Riscos, J. A. Alonso, E. Briales, Multivalued logic and Gröbner Bases with applications to modal logic, *J. Symb. Comput.*, **11** (1991), 181–194. https://doi.org/10.1016/S0747-7171(08)80043-0
5. 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.
6. 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. Comput. Simul.*, **45** (1998), 83–99.
7. 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, *J. Symbol. Comput.*, **41** (2006), 475–511. https://doi.org/10.1016/j.jsc.2005.09.007
8. 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.
9. J. Abbott, A. M. Bigatti, CoCoALib: a C++ library for doing Computations in Commutative Algebra, 2019. Available from: http://cocoa.dima.unige.it/cocoalib
10. 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, *Comput. Math. Appl.*, **51** (2000), 473–481.

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, *Math. Comput. Simul.*, **58** (2002), 183–202.
12. 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. https://doi.org/10.1016/j.eswa.2005.09.011
13. M. Lourdes Jimenez, J. M. Santamaría, R. Barchino, L. Laita, L. M. Laita, L. A. González, et al., Knowledge representation for diagnosis of care problems through an expert system: Model of the auto-care deficit situations, *Expert Syst. Appl.*, **34** (2008), 2847–2857. https://doi.org/10.1016/j.eswa.2007.05.039
14. E. Roanes-Lozano, J. L. Galán-García, G. Aguilera-Venegas, A prototype of a RBES for personalized menus generation, *Appl. Math. Comput.*, **315** (2017), 615–624. https://doi.org/10.1016/j.amc.2016.12.023
15. G. Aguilera-Venegas, E. Roanes-Lozano, E. Rojo-Martínez, J. L. Galán-García, A proposal of a mixed diagnostic system based on decision trees and probabilistic experts rules, *J. Comput. Appl. Math.*, **427** (2023). https://doi.org/10.1016/j.cam.2023.115130
16. G. Aguilera-Venegas, A. López-Molina, G. Rojo-Martínez, J. L. Galán-García, Comparing and tuning machine learning algorithms to predict type 2 diabetes mellitus, *J. Comput. Appl. Math.*, **427** (2023). https://doi.org/10.1016/j.cam.2023.115115
17. E. Roanes-Lozano, E. A. Casella, F. Sánchez, A. Hernando, Diagnosis in Tennis Serving Technique, *Algorithms*, **13** (2020). https://doi.org/10.3390/a13050106
18. A. Roanes-Lozano, J. L. Galán-García, G. Aguilera-Venegas, A prototype of a functional approach to personalized menus generation using set operations, *Adv. Comput. Math.*, **13** (2019), 1881–1895.
19. M. Villalba-Orero, E. Roanes-Lozano, A prototype of a decision support system for equine cardiovascular diseases diagnosis and management, *Mathematics*, **20** (2021). https://doi.org/10.3390/math9202580
20. A. Hernando, A new algebraic model for implementing expert systems represented under the 'Concept-Attribute-Value' paradigm, *Math. Comput. Simul.*, **82** (2011), 29–43. https://doi.org/10.1016/j.matcom.2010.06.020
21. A. Hernando, R. Maestre-Martínez, E. Roanes-Lozano, A natural language for implementing algebracially expert systems, *Math. Comput. Simul.*, **129** (2016), 31–49. https://doi.org/10.1016/j.matcom.2016.04.006
22. M. Brickenstein, A. DreyerPolyBoRi: Polybori: A framework for Gröbner-basis computations with Boolean polynomials, Journal of Symbolic Computation, **44** (2009), 1326–1345. https://doi.org/10.1016/j.jsc.2008.02.017
23. E. Roanes-Lozano, L. M. Laita, An applicable topology-independent model for railway interlocking systems, *Math. Comput. Simul.*, **45** (1998), 175–183. https://doi.org/10.1016/S0378-4754(97)00093-1
24. E. Roanes-Lozano, L. M. Laita, E. Roanes-Macías, An application of an AI methodology to railway interlocking systems using computer algebra, in *Tasks and Methods in Applied Artificial Intelligence*, (1998), 687–696. https://doi.org/10.1007/3-540-64574-8_455
25. E. Roanes-Lozano, E. Roanes-Macías, L. Laita, Railway interlocking systems and Gröbner bases, *Math. Comput. Simul.*, **51** (2000), 473–481. https://doi.org/10.1016/S0378-4754(99)00137-8

26. E. Roanes-Lozano, A. Hernando, J. A. Alonso, L. M. Laita, A logic approach to decision taking in a railway interlocking system using Maple, *Math. Comput. Simul.*, **82** (2011), 15–28. https://doi.org/10.1016/j.matcom.2010.05.024
27. A. Hernando, E. Roanes-Lozano, R. Maestre-Martínez, J. Tejedor, A logic-algebraic approach to decision taking in a railway interlocking system, *Ann. Math. Artif. Intell.*, **65** (2012), 317–328. https://doi.org/10.1007/s10472-012-9321-y
28. A. Hernando, R. Maestre, E. Roanes-Lozano, A new algebraic approach to decision making in a railway interlocking system based on preprocess, *Math. Probl. Eng.*, **2018** (2018), 4982974. https://doi.org/10.1155/2018/4982974
29. A. Hernando, E. Roanes-Lozano, J. L. Galán-García, G. Aguilera-Venegas, Decision making in railway interlocking systems based on calculating the remainder of dividing a polynomial by a set of polynomials, *Electron. Res. Arch.*, **31** (2023), 6160–6196. https://doi.org/10.3934/era.2023313