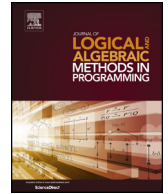




Contents lists available at ScienceDirect

# Journal of Logical and Algebraic Methods in Programming

[www.elsevier.com/locate/jlamp](http://www.elsevier.com/locate/jlamp)


## Inference engine based on closure and join operators over Truth Table Binary Relations



Samir Elloumi<sup>a</sup>, Bilel Boulifa<sup>a</sup>, Ali Jaoua<sup>a,\*</sup>, Mohammad Saleh<sup>a</sup>,  
Jameela Al Otaibi<sup>a</sup>, Marcelo F. Frias<sup>b</sup>

<sup>a</sup> Department of Computer Science, Qatar University, Qatar

<sup>b</sup> Department of Software Engineering, Instituto Tecnológico de Buenos Aires, CONICET, Argentina

### ARTICLE INFO

#### Article history:

Available online 12 February 2014

#### Keywords:

Truth Table Binary Relation  
Relation combining  
Closure operators  
Formal concept analysis  
Inference engine  
Cooperative conceptual reasoning

### ABSTRACT

We propose a conceptual reasoning method for an inference engine. Starting from a knowledge base made of decision rules, we first map each rule to its corresponding Truth Table Binary Relation (TTBR), considered as a formal context. Objects in the domain of TTBR correspond to all possible rule interpretations (in terms of their truth value assignments), and elements in the range of TTBR correspond to the attributes. By using the 'natural join' operator in the 'ContextCombine' Algorithm, we combine all truth tables into a global relation which has the advantage of containing the complete knowledge of all deducible rules. By conceptual reasoning using closure operators, from the initial rules we obtain all possible conclusions with respect to the global relation. We may then check if expected goals are among these possible conclusions. We also provide an approximate solution for the exponential growth of the global relation, by proposing modular and cooperative conceptual reasoning. We finally present experimental results for two case studies and discuss the effectiveness of our approach.

© 2014 Elsevier Inc. All rights reserved.

### 1. Introduction

Automated reasoning has applications in domains such as automated theorem proving [1], software verification [2] and model finding [3]. Automated reasoning also plays an important role in expert systems [4,5], where techniques such as forward chaining, backward chaining, mixed or structural approaches are employed [6]. In this paper, for each decision rule involving some set of terms or attributes  $P$ , we create its truth table as a binary relation  $R$  over all involved attributes. We then combine all tables in a single one using our proposed 'ContextCombine' Algorithm. The latter explores all possibilities between the different assignments to generate a global solution in a new truth table. By using a Galois connection on the entire table thus obtained, we are able to infer all possible conclusions related with some input facts. Furthermore, we are able to regenerate all implications from this new context [7]. This procedure may produce tables whose size grows beyond reasonable limits. To cope with this limitation of the technique we propose a modular and cooperative reasoning approach that delays combination operations, by first reducing the different initial contexts associated with the different rules with respect to the facts initially submitted to the inference engine.

The article is organized as follows. In Section 2 we present the foundations of formal concept analysis. In Section 3 we propose *conceptual reasoning*, a reasoning method based on Galois connection. In Section 4 we propose a modular

\* Corresponding author.

E-mail addresses: [eloumis@qu.edu.qa](mailto:eloumis@qu.edu.qa) (S. Elloumi), [boulifa.bilel@gmail.com](mailto:boulifa.bilel@gmail.com) (B. Boulifa), [jaoua@qu.edu.qa](mailto:jaoua@qu.edu.qa) (A. Jaoua), [Mohd.saleh@qu.edu.qa](mailto:Mohd.saleh@qu.edu.qa) (M. Saleh), [jameelaa@gmail.com](mailto:jameelaa@gmail.com) (J. Al Otaibi), [mfras@itba.edu.ar](mailto:mfras@itba.edu.ar) (M.F. Frias).

<http://dx.doi.org/10.1016/j.jlap.2014.02.007>

2352-2208/© 2014 Elsevier Inc. All rights reserved.

and cooperative reasoning approach which allows us to partially overcome the limitations of conceptual reasoning. Initial experimentation using this approach shows its efficiency in the two case studies (cf. Section 5.1 and Section 5.2) related to SAT/UNSAT problems and Medical Data, respectively. Finally, in Section 6, we draw some conclusions and present some proposals for further work.

## 2. Formal Concept Analysis and Relational Algebra

We first recall some basic notions from Formal Concept Analysis (FCA) [8,9] and Relational Algebra [10].

**Definition 1.** Let  $\mathcal{O}$  and  $\mathcal{P}$  be sets, called the set of objects and attributes, respectively. Let  $\mathcal{R}$  be a relation on  $\mathcal{O} \times \mathcal{P}$ . For  $o \in \mathcal{O}$  and  $p \in \mathcal{P}$ ,  $\mathcal{R}(o, p)$  holds if the object  $o$  has attribute  $p$ , denoted also by  $(o, p) \in \mathcal{R}$ . The triple  $\mathcal{K} = (\mathcal{O}; \mathcal{P}; \mathcal{R})$  is called a *formal context*.

We may notice that the definition of a formal context is very similar to a relation where  $\mathcal{O}$  (respectively,  $\mathcal{P}$ ) represents the domain (respectively, the range) of the relation.

### 2.1. Galois connections and their properties

**Definition 2.** Let  $(A, \leq_A)$  and  $(B, \leq_B)$  be two partially ordered sets. Let  $f : A \rightarrow B$  and  $g : B \rightarrow A$  such that  $\forall a \in A, b \in B$ ,  $f(a) \leq_B b \iff g(b) \leq_A a$ . Then, the pair  $(f, g)$  is called a *Galois connection*.

**Proposition 3.** Let  $\mathcal{O}, \mathcal{P}$  be two arbitrary sets. Let  $\mathcal{R}$  be a relation on  $\mathcal{O} \times \mathcal{P}$ . Let  $A \subseteq \mathcal{O}$  and  $B \subseteq \mathcal{P}$  also be arbitrary. The pair of functions  $(f, g)$  with  $f : 2^{\mathcal{O}} \rightarrow 2^{\mathcal{P}}$  and  $g : 2^{\mathcal{P}} \rightarrow 2^{\mathcal{O}}$  defined by

- $f(A) = \{p \in \mathcal{P} \mid \forall o \in A, (o, p) \in \mathcal{R}\}$ ,
- $g(B) = \{o \in \mathcal{O} \mid \forall p \in B, (o, p) \in \mathcal{R}\}$ ,

forms a Galois connection.

Let  $A_1, A_2 \subseteq \mathcal{O}$  and  $B_1, B_2 \subseteq \mathcal{P}$ . It is well known [9] that a pair  $(f, g)$  forms a Galois connection if and only if the following properties are satisfied:

$$\begin{aligned} A_1 \subseteq A_2 &\Rightarrow f(A_2) \subseteq f(A_1), & B_1 \subseteq B_2 &\Rightarrow g(B_2) \subseteq g(B_1), \\ A \subseteq (g \circ f)(A), & & B &\subseteq (f \circ g)(B). \end{aligned}$$

**Definition 4.** We call  $(g \circ f)(A)$  the *closure* of  $A$ , and  $(f \circ g)(B)$  the *closure* of  $B$ . The pair  $(A, B)$ , where  $A$  is included in  $\mathcal{O}$ ,  $B$  is included in  $\mathcal{P}$ ,  $f(A) = B$ , and  $g(B) = A$  is called a *formal concept* of context  $\mathcal{K}$  with extent  $A$  and intent  $B$ . We also have  $(g \circ f)(A) = A$  and  $(f \circ g)(B) = B$ .

### 2.2. Rule representation and reasoning

We are mainly concerned with knowledge base representation and automated reasoning. In this section we show how a truth table associated with a decision rule is represented as a formal context, and how different rules may be combined if there is no contradiction between them.

#### 2.2.1. Rule representation

A decision rule of the form ‘if  $A$  then  $B$ ’ or equivalently ‘ $A \rightarrow B$  is true’, between attributes  $A$  and  $B$ , reflects the knowledge of a given domain in which the satisfaction of premise  $A$  implies the conclusion  $B$ . From a logical point of view, the expression ‘ $A \rightarrow B$  is true’ can be represented as a Truth Table (TT) where each row (or solution) is a truth value assignment ( $1 = \text{true}, 0 = \text{false}$ ) for the attributes  $A$  and  $B$ . For instance, the assignment  $(A = 0, B = 1)$  is a solution for ‘ $A \rightarrow B$  is true’. At the same time, the later expression could be also represented as a Formal Context (FC)  $\mathcal{K} = (\mathcal{O}; \mathcal{P}; \mathcal{R})$ , where the set  $\mathcal{O}$  is the set of possible solutions (truth-value assignments for  $A$  and  $B$  such that  $A \rightarrow B$  is true), the set  $\mathcal{P}$  is the set of attributes (or properties, in our case  $\mathcal{P} = \{A, B\}$ ), and  $\mathcal{R}(o, p)$  holds if the solution  $o \in \mathcal{O}$  has the assignment *true* for the attribute  $p \in \mathcal{P}$ . In the remainder, the Truth Table Binary Relation  $\mathcal{R}$  will be denoted by TTBR and the expression ‘ $A \rightarrow B$  is true’ will be abbreviated as ‘ $A \rightarrow B$ ’.

**Example 5.** Let us consider the rule  $A \rightarrow B$ . In Table 1 we find a representation of a formal context  $\mathcal{K} = (\mathcal{O}; \mathcal{P}; \mathcal{R})$ , where  $\mathcal{O} = \{s_1, s_2, s_3\}$ ,  $\mathcal{P} = \{A, B\}$  and  $\mathcal{R} = \{(s_1, A), (s_1, B), (s_2, B)\}$ .

**Table 1**  
TTBR<sub>1</sub> for rule  $A \rightarrow B$ .

	A	B
$s_1$	1	1
$s_2$	0	1
$s_3$	0	0

**Table 2**  
TTBR<sub>2</sub> for rule IF (B OR D) THEN K.

	B	D	K
$s'_1$	0	1	1
$s'_2$	1	0	1
$s'_3$	0	0	0
$s'_4$	0	0	1
$s'_5$	1	1	1

**Remark 6.** Notice that  $(f \circ g)(\{A\}) = \{A, B\}$ , which means that the closure of set  $\{A\}$  is  $\{A, B\}$ ; which is interpreted as  $A \rightarrow B$ . We may also conclude that  $A \rightarrow B$  because  $g(\{A\}) = \{s_1\}$  is included in  $g(\{B\}) = \{s_1, s_2\}$ . According to Guigues et al. [7], the implication  $A \rightarrow B$  holds if  $g(\{A\}) \subseteq g(\{B\})$ .

**Remark 7.** Formally, we get for arbitrary sets  $\mathcal{O}$  of objects and  $\mathcal{B}$  of attributes, and TTBR  $\mathcal{R}$ ,

$$Y \in (f \circ g)(\mathcal{B}) \iff \forall s \in \mathcal{O} : (\forall X \in \mathcal{B} : \mathcal{R}(s, X)) \rightarrow \mathcal{R}(s, Y).$$

In words:  $(f \circ g)(\mathcal{B})$  consists of all those attributes  $Y$  which are logically implied by the attributes  $X$  in  $\mathcal{B}$  with respect to  $\mathcal{R}$ .

In case of a decision rule having logical connectors AND, OR, NOT, we apply the same principle by enumerating all possible solutions and representing them as a TTBR. An example follows.

**Example 8.** Let us consider rule IF (B OR D) THEN K. In Table 2 we find a representation of a context  $\mathcal{K} = (\mathcal{O}; \mathcal{P}; \mathcal{R})$  where:

- $\mathcal{O} = \{s'_1, s'_2, s'_3, s'_4, s'_5\}$ ,
- $\mathcal{P} = \{B, D, K\}$ , and
- $\mathcal{R} = \{(s'_1, D), (s'_1, K), (s'_2, B), (s'_2, K), (s'_4, K), (s'_5, B), (s'_5, D), (s'_5, K)\}$ .

### 2.2.2. Context combination

In order to represent the logic behind a set of decision rules we propose to combine the corresponding TTBRs into a single TTBR. Once we obtain the global TTBR we can start the reasoning process as explained in Section 3. The combining procedure is akin to the equi-join operator in relational database theory [11]. Combining two TTBRs, let's say  $TTBR_1$  and  $TTBR_2$ , consists of processing case by case the consistency of any row in  $TTBR_1$  with any other row in  $TTBR_2$ . If not found contradicting, the two checked rows are combined into the final TTBR. Otherwise, they will be discarded. So, by definition, the obtained combined relation is simultaneously consistent with  $TTBR_1$  and  $TTBR_2$ . If two relations  $TTBR'_1$  and  $TTBR'_2$  are not sharing any attribute, then any row in  $TTBR'_1$  is combined to any other row in  $TTBR'_2$  in the target relation.

**Example 9.** If we consider a knowledge base composed of the rules presented in Examples 5 and 8, we can combine their associated TTBRs as shown in Table 3. Combining the two different truth tables consists of producing a new one while preserving the coherence between them (i.e., the same truth values assignment for the common attributes). Fig. 1 illustrates how  $TTBR_1$  and  $TTBR_2$  are combined in order to get  $TTBR_3$  (shown in Table 3).

Algorithm 1 describes how to combine two TTBRs while preserving the coherent information that they share. In case of independent rules, i.e., rules with disjoint sets of attributes (see Example 10), all table's solutions are combined by computing their Cartesian product (lines 4–5). Otherwise, we retain only rows having the same truth values (designated by function 'SameSolution') for their common attributes (line 9). The addition of a new row in table  $\mathcal{R}_3$  is done by procedure 'createNewObject' (line 10), that works as follows: let  $TV(s, p, R)$  be the truth value assigned to attribute  $p$  in the solution  $s$  within relation  $R$ .  $TV(s, p, R)$  is assigned 1 if  $R(s, p) = 1$ ; otherwise  $TV(s, p, R)$  is assigned 0. Hence, for all  $s_k$  in  $O_3$ ,  $s_i$  in  $O_1$ ,  $s_j$  in  $O_2$ , the following properties should hold:

- $TV(s_k, p, R_3) = TV(s_i, p, R_1) = TV(s_j, p, R_2)$  for all  $p$  in  $CA$ ,
- $TV(s_k, p, R_3) = TV(s_i, p, R_1)$  for all  $p$  in  $P_1$ ,
- $TV(s_k, p, R_3) = TV(s_i, p, R_2)$  for all  $p$  in  $P_2$ .

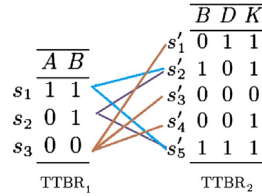


Fig. 1. How the combining is done.

Table 3  
TTBR<sub>3</sub>, result of combining TTBR<sub>1</sub> and TTBR<sub>2</sub>.

	A	B	D	K
cs <sub>1</sub>	0	0	1	1
cs <sub>2</sub>	1	1	0	1
cs <sub>3</sub>	0	1	0	1
cs <sub>4</sub>	0	0	0	0
cs <sub>5</sub>	0	0	0	1
cs <sub>6</sub>	0	1	1	1
cs <sub>7</sub>	1	1	1	1

Table 4  
The combining of R<sub>1</sub> and R<sub>2</sub>.

	A	B	C	D
cs <sub>1</sub>	0	1	1	1
cs <sub>2</sub>	0	0	1	1
cs <sub>3</sub>	1	1	0	1
cs <sub>4</sub>	1	1	0	0
cs <sub>5</sub>	0	0	0	0
cs <sub>6</sub>	1	1	1	1
cs <sub>7</sub>	0	1	0	1
cs <sub>8</sub>	0	1	0	0
cs <sub>9</sub>	0	0	0	1

**Algorithm 1:** TwoContextsCombination.

```

Input:  $\mathcal{K}_1 = (\mathcal{O}_1; \mathcal{P}_1; \mathcal{R}_1)$ ,  $\mathcal{K}_2 = (\mathcal{O}_2; \mathcal{P}_2; \mathcal{R}_2)$ : two TTBRs
Output:  $\mathcal{K}_3 = (\mathcal{O}_3; \mathcal{P}_3; \mathcal{R}_3)$ : combined TTBR
1 begin
2    $\mathcal{P}_3 \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2$  // new properties in the new context  $\mathcal{K}_3$ 
3    $CA \leftarrow \mathcal{P}_1 \cap \mathcal{P}_2$  // identify the common attributes.
4   if  $(CA = \emptyset)$  then
5      $\mathcal{R}_3 = \text{CartesianProduct}(\mathcal{R}_1, \mathcal{R}_2)$ 
6   else
7     foreach Solution  $s_i \in \mathcal{O}_1$  do
8       foreach Solution  $s_j \in \mathcal{O}_2$  do
9         if SameSolution( $s_i, s_j, CA$ ) then // same truth values for CA in  $s_i$  and  $s_j$ 
10          createNewObject( $s_i, s_j, \mathcal{O}_3, \mathcal{R}_3$ ) // we add new solution  $s_k$  to  $\mathcal{O}_3$  such that:
11           $\mathcal{R}_3(s_k, p) = \mathcal{R}_1(s_i, p) = \mathcal{R}_2(s_j, p), \forall p \in CA$ 
12           $\mathcal{R}_3(s_k, p) = \mathcal{R}_1(s_i, p) = \forall p \in \mathcal{P}_1$ 
13           $\mathcal{R}_3(s_k, p) = \mathcal{R}_2(s_j, p) = \forall p \in \mathcal{P}_2$ 

```

**Example 10.** Suppose that the knowledge base is composed of two independent decision rules:  $R_1 = A \rightarrow B$  and  $R_2 = C \rightarrow D$ . Then, as a first step, we create two TTBRs equivalent to those in Table 1. Algorithm 1 combines the formal contexts associated with  $R_1$  and  $R_2$ , and produces the TTBR depicted in Table 4.

The algorithm's worst-case running time complexity is in  $O(|\mathcal{O}_1| \times |\mathcal{O}_2| \times |CA|)$ , where  $|S|$  denotes the cardinality of set  $S$ .

Once the global TTBR has been obtained, we can start the reasoning process using the closure operator. In fact, from Table 4, we calculate  $(f \circ g)(\{A, C\}) = \{A, C, B, D\}$ , which means that  $(A \wedge C) \rightarrow (B \wedge D)$ . We can also infer other rules, such as  $(A \wedge D) \rightarrow B$ , because  $(f \circ g)(\{A, D\}) = \{A, B, D\}$ .

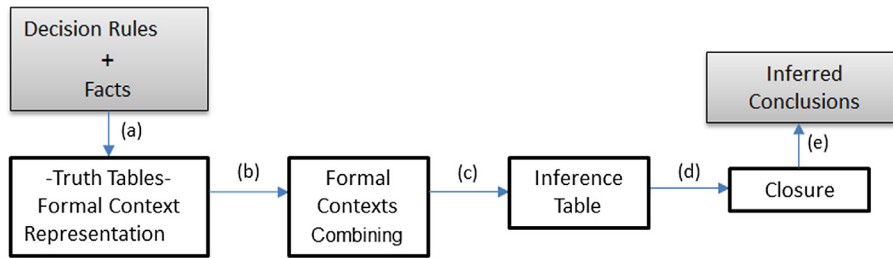


Fig. 2. A schematic description of conceptual reasoning.

Table 5  
Example of knowledge base.

Id	Rule
1	IF A THEN B
2	IF C THEN D
3	IF M THEN E
4	IF K THEN F
5	IF G THEN H
6	IF I THEN J
7	IF B OR D THEN K
8	IF E THEN L
9	IF (F AND H) OR J THEN M
10	IF K AND L THEN N
11	IF M THEN O
12	IF N OR O THEN P

Also, from Table 3 we obtain  $(f \circ g)(\{A\}) = \{A, B, K\}$  which indicates that  $A$  implies both  $B$  and  $K$ , which is coherent with respect to the considered rules.

### 3. Conceptual reasoning

In this section we detail our new approach based on conceptual reasoning as illustrated in Fig. 2.

#### 3.1. General description

Starting from a set of rules in a knowledge based system, we generate, for each decision rule, a TTBR as indicated in Fig. 2(a). Then, combining the two contexts as per Algorithm 1 (cf. Fig. 2(b)), we obtain a total relation reflecting all possible cases (or solutions) which are equivalent to the initial set of decision rules (see Fig. 2(c)). The advantage of this method is that conflicts are detected during the combining step. In fact, if two or more rules are in conflict we obtain an empty TTBR. In that case, the user is aware about the inconsistency of the knowledge base rules. After this step, starting from the initial fact  $A$ , and applying the Galois connection, the expression  $(f \circ g)(\{A\}) - \{A\}$  gives us the set of all deduced attributes (see Fig. 2(d)–(e)). In this article we have also used, as an additional reasoning option, the Duquenne–Guigues set of implications extracted using the ConImp tool [12,13]. Hence, we could get all the deduced attributes from the implications without using the closure operator. The tool also allows the removal of redundancies from the initial relation.

**Example 11.** Let us consider the knowledge base depicted in Table 5, composed of 12 rules using 16 attributes:  $A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P$ . The TTBR combining the 12 rules contains 258 rows. We present a reduced subset in Table 6.

To illustrate the case of possible inconsistency that we could detect by TTBR-combining, and for the sake of illustration, we consider rule  $r_{13} = \text{NOT}(\text{IF } A \text{ THEN } B)$  to be added to the knowledge base depicted in Table 5. The unique solution satisfying  $r_{13}$  is  $s_1 = \{A = 1, B = 0\}$ . Obviously, this rule is logically in conflict with rule  $r_1 = (\text{IF } A \text{ THEN } B)$  and the Global Context will be empty. It could be interesting for the user in such cases to revise the knowledge base and remove the possible inconsistency.

#### 3.2. Reasoning

After producing the whole non-empty TTBR according to the knowledge base rules, we can start the reasoning step (see Fig. 2(d)–(e)). Reasoning, or deductive reasoning [14], means the inference of possible conclusions from valid facts. In

**Table 6**

Subset of the Global Context obtained from the knowledge base in Table 5.

	A	B	C	D	M	E	K	F	G	H	I	J	L	N	O	P
$s_1$	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1
$s_2$	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1
$s_3$	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0
$s_4$	0	0	0	0	0	1	0	0	1	1	0	0	1	1	0	1
$s_5$	0	0	0	0	0	1	0	0	1	1	0	0	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
$s_{258}$	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1

**Algorithm 2:** Deductive reasoning.

```

Input:

- $\mathcal{K} = (\mathcal{O}; \mathcal{P}; \mathcal{R})$ : TTBR representing combined Truth Tables
- $DGI$ : Duquenne–Guigues set of implications extracted from  $\mathcal{K}$
- $Exp$ : Boolean expression representing the initial facts.

Output:

- $Conc$ : Derived conclusions

1 begin
2   if  $Exp \in DGI$  then
3      $Conc \leftarrow \text{ExtractDirectConclusion}(Exp, DGI)$ 
4   else
5     if  $AllConjunctive(Exp)$  then
6       //All attributes in  $Exp$  are connected with logical AND operator.
7       //Hence, we get new conclusions by Galois Connection.
8        $Conc \leftarrow (f \circ g)(cToSet(Exp))$ 
9     else
10       $NewAttrib \leftarrow \text{EvaluateLogicExpression}(Exp)$ 
11      //  $NewAttrib$  is a new column to be added to  $\mathcal{K}$ .
12      //  $\forall o \in \mathcal{O}, \mathcal{R}(o, NewAttrib) = 1$  if  $Exp$  is true for  $o$  and 0 otherwise.
13       $\mathcal{K} \leftarrow \text{AddAttrib}(NewAttrib, \mathcal{K})$ 
14       $Conc \leftarrow (f \circ g)(\{NewAttrib\})$ 

```

Algorithm 2 the valid facts are represented by parameter  $Exp$ , which may contain a single fact or a combination of facts obtained using logical connectors. The details are explained in the following paragraphs.

- If  $Exp$  represents a single fact (i.e., an attribute assumed to be true as input, let's say  $A$ ): We can derive the possible conclusions by applying the closure operator. But we may instead, as indicated in lines 2–3, directly retrieve the Duquenne–Guigues set of conclusions using the ConImp tool [12,13]. An example of such rules is presented in Table 7. We observe all the possible conclusions that may be derived from the initial facts. For example, in row 3, we observe that properties  $D$ ,  $K$  and  $F$  are derived from fact  $C$ .
- In case of an  $Exp$  representing a conjunction of attributes (line 5), let's say  $(A \wedge B \wedge C \dots)$ : We compute the conclusions by applying the closure operator to the set representing  $Exp$  (line 8), i.e.,  $\{A, B, C, \dots\}$ . The function  $cToSet()$  (line 8) converts  $Exp$  to a set.
- In case of an  $Exp$  representing an arbitrary Boolean combination of attributes, let's say  $((A \vee B) \wedge \neg C \dots)$ : We have to perform the following steps:
  1. Add a new attribute 'NewAttrib' reflecting  $Exp$  and compute its truth-value for all solutions in TTBR. This is done by procedures 'EvaluateLogicExpression' and 'NewAddAttrib' (lines 10–13).
  2. Derive possible conclusions by applying the closure operator to  $\{NewAddAttrib\}$  (line 14).

In the end, a propositional logic rule may be mapped to an expression and vice versa:  $A \rightarrow B$  is equivalent to  $(NOT B \vee A)$ . So in our case expressions and rules are all processed in the same way. However, in most expert systems, a knowledge data base may be presented with a set of rules using or not negations. The extension we make in this article for handling negation (see Section 3.3) makes it possible to process any kind of Boolean expression.

The algorithm complexity is closely related to the size of the obtained TTBR, i.e., it depends on  $|\mathcal{O}|$ ,  $|Exp|$  (the number of attributes in  $Exp$ ), as well as on the number  $NDGI$  of Duquenne–Guigues implications. The worst-case running time complexity is in  $O(\max(NDGI, |\mathcal{O}|) \times |Exp|)$ .

The characteristic of the reasoning process based on the Duquenne–Guigues set of implications, as well as on Galois connection, is to derive positive conclusions according to the considered TTBR. In reality, negated attributes could also be deduced by expanding the initial TTBR as follows: For each attribute  $A$  we add to the TTBR its negated version, i.e.,  $\neg A$ . Obviously, for all  $o \in \mathcal{O}$ , the truth value  $TV(o, A, R)$  equals  $1 - TV(o, \neg A, R)$ . Hence, we get an entire duplication of the

**Table 7**  
Duquenne–Guigues set of implications discovered from the reduced TTBR.

Id	Implication
1	$A \rightarrow B K F$
2	$B \rightarrow K F$
3	$C \rightarrow D K F$
4	$D \rightarrow K F$
5	$E \rightarrow L$
6	$K \rightarrow F$
7	$G \rightarrow H$
8	$M \rightarrow E L O P$
9	$F H \rightarrow M E L O P$
10	$I \rightarrow M E J L O P$
11	$J \rightarrow M E L O P$
12	$K F L \rightarrow N P$
13	$N \rightarrow P$
14	$O \rightarrow P$

initial TTBR. In Section 3.3 we propose a new solution to avoid such duplication by extending the Galois connection. In the following we present the new definitions for the extended Galois connection, as well as its main properties and their proofs.

### 3.3. Extended Galois connection

In this subsection we propose a novel solution for the reasoning process, that consists of extending the Galois connection to derive both positive and negative conclusions. Let  $\mathcal{K} = (\mathcal{O}; \mathcal{P}; \mathcal{R})$  be a formal context,  $A \subseteq \mathcal{O}$  and  $B \subseteq \mathcal{P}$ . We extend the functions  $(f, g)$ , defined in Section 2.1, by  $f_e$  and  $g_e$  respectively to process negative attributes, as shown:

- $f_e(A) = \{p \in \mathcal{P} \mid (o, p) \in \mathcal{R}, \forall o \in A\} \cup \{\neg p \mid p \in \mathcal{P} \wedge (o, p) \notin \mathcal{R}, \forall o \in A\}$ ,
- $g_e(B) = \{o \in \mathcal{O} \mid (o, p) \in \mathcal{R} \text{ if } p \text{ positive and } (o, \neg p) \notin \mathcal{R} \text{ if } p \text{ negative, } \forall p \in B\}$ .

**Example 12.** Let us consider the TTBR from Table 1. We have the following results:

- $f_e(\{s_2, s_3\}) = \{\neg A\}$ ,
- $f_e(\{s_1\}) = \{A, B\}$ ,
- $f_e(\{s_1, s_2\}) = \{B\}$ ,
- $g_e(\{\neg A\}) = \{s_2, s_3\}$ ,
- $g_e(\{\neg B\}) = \{s_3\}$ ,
- $f_e(\{s_1, s_2, s_3\}) = \{\}$ .

**Lemma 13.** Let  $\mathcal{K} = (\mathcal{O}; \mathcal{P}; \mathcal{R})$  be a formal context. Functions  $f_e$  and  $g_e$  form a Galois connection between the powersets of  $\mathcal{O}$  and  $\mathcal{P}$ .

**Proof.** Let  $A \subseteq \mathcal{O}$ ,  $A_1 \subseteq A_2 \subseteq \mathcal{O}$ ,  $B \subseteq \mathcal{P}$ , and  $B_1 \subseteq B_2 \subseteq \mathcal{P}$ . According to Proposition 3, we will prove that the following properties are satisfied:

- (I)  $A_1 \subseteq A_2 \Rightarrow f_e(A_2) \subseteq f_e(A_1)$ ,      (II)  $B_1 \subseteq B_2 \Rightarrow g_e(B_2) \subseteq g_e(B_1)$ ,  
 (III)  $A \subseteq (g_e \circ f_e)(A)$ ,      (IV)  $B \subseteq (f_e \circ g_e)(B)$ .

- (I)  $A_1 \subseteq A_2 \Rightarrow f_e(A_2) \subseteq f_e(A_1)$ :  
 $f_e(A_2) = \{p \in \mathcal{P} \mid (o, p) \in \mathcal{R}, \forall o \in A_2\} \cup \{\neg p \mid p \in \mathcal{P} \wedge (o, p) \notin \mathcal{R}, \forall o \in A_2\}$ . Since  $A_1 \subseteq A_2$ , all the conditions satisfied by the elements of  $A_2$  are necessarily satisfied by all its members, in particular for all elements from  $A_1$ . Then,

$$\begin{aligned} f_e(A_2) &= \{p \in \mathcal{P} \mid (o, p) \in \mathcal{R}, \forall o \in A_2\} \cup \{\neg p \mid p \in \mathcal{P} \wedge (o, p) \notin \mathcal{R}, \forall o \in A_2\} \\ &\subseteq \{p \in \mathcal{P} \mid (o, p) \in \mathcal{R}, \forall o \in A_1\} \cup \{\neg p \mid p \in \mathcal{P} \wedge (o, p) \notin \mathcal{R}, \forall o \in A_1\} \\ &\subseteq f_e(A_1). \end{aligned}$$

- (II)  $B_1 \subseteq B_2 \Rightarrow g_e(B_2) \subseteq g_e(B_1)$ :

$$g_e(B_2) = \{o \in \mathcal{O} \mid (o, p) \in \mathcal{R} \text{ if } p \text{ positive and } (o, \neg p) \notin \mathcal{R} \text{ if } p \text{ negative, } \forall p \in B_2\}.$$

Since the definition is preserved by subsets of  $B_2$ , the proof follows the steps of the previous case.

- (III)  $A \subseteq (g_e \circ f_e)(A)$ :

By definition of  $f_e$ ,

$$f_e(A) = \{p \in \mathcal{P} \mid (o, p) \in \mathcal{R}, \forall o \in A\} \cup \{\neg p \mid p \in \mathcal{P} \wedge (o, p) \notin \mathcal{R}, \forall o \in A\}. \quad (1)$$

By definition of  $g_e$ ,

$$(g_e \circ f_e)(A) = \{o \in \mathcal{O} \mid (o, p) \in \mathcal{R} \text{ if } p \text{ positive and } (o, \neg p) \notin \mathcal{R} \text{ if } p \text{ negative, } \forall p \in f_e(A)\}. \quad (2)$$

Let us assume there exists  $o \in A$  such that  $o \notin (g_e \circ f_e)(A)$ , and let us look for a contradiction. Two possibilities arise, namely, there exists  $p \in f_e(A)$  such that  $(o, p) \notin \mathcal{R}$  with  $p$  positive, or there exists  $p \in f_e(A)$  such that  $(o, \neg p) \in \mathcal{R}$  with  $p$  negative. Let us analyze these two cases:

– There exists  $p \in f_e(A)$  such that  $(o, p) \notin \mathcal{R}$  with  $p$  positive:

Since  $p$  is a positive literal, according to (1), it must satisfy  $(o, p) \in \mathcal{R}$ , which leads to a contradiction.

– There exists  $p \in f_e(A)$  such that  $(o, \neg p) \in \mathcal{R}$  with  $p$  a negative literal:

Since  $p$  is negative,  $p = \neg q$  for some attribute  $q$ . According to (1)  $\neg q$  must belong to the term on the right-hand side of the union. Therefore,  $(o, q) \notin \mathcal{R}$ , which is to say that  $(o, \neg p) \notin \mathcal{R}$ , which leads to a contradiction.

- (IV)  $B \subseteq (f_e \circ g_e)(B)$ :

By definition of  $g_e$ ,

$$g_e(B) = \{o \in \mathcal{O} \mid \underbrace{(o, p) \in \mathcal{R} \text{ if } p \text{ positive}}_{LHS} \text{ and } \underbrace{(o, \neg p) \notin \mathcal{R} \text{ if } p \text{ negative}}_{RHS}, \forall p \in B\}. \quad (3)$$

By definition of  $f_e$ ,

$$(f_e \circ g_e)(B) = \{p \in \mathcal{P} \mid (o, p) \in \mathcal{R}, \forall o \in g_e(B)\} \cup \{\neg p \mid p \in \mathcal{P} \wedge (o, p) \notin \mathcal{R}, \forall o \in g_e(B)\}. \quad (4)$$

Let us assume there is  $p \in B$  such that  $p \notin (f_e \circ g_e)(B)$ , and let us arrive at a contradiction. If  $p$  is positive, according to Eq. (4) there exists  $o \in g_e(B)$  such that  $(o, p) \notin \mathcal{R}$ . But this contradicts term *LHS* in Eq. (3). Similarly, if  $p$  is negative, there exists  $o_2 \in g_e(B)$  such that  $(o, \neg p) \in \mathcal{R}$ . But this now contradicts term *RHS* in Eq. (3).  $\square$

So far, conceptual reasoning considers the global TTBR to ensure the consistence of the knowledge base, as well as the derivation of any new positive or negative conclusion. However, a serious difficulty may arise when the size of the TTBR increases considerably. The truth table associated with  $N$  properties may contain up to  $2^N$  solutions. To tame this exponential growth in size, we propose to delay the construction of the global TTBR by reducing the different TTBRs during the reasoning step. We detail this idea in the following section.

#### 4. Cooperative and conceptual reasoning

Starting from the methods proposed in [15], we can make the reasoning effort modular and cooperative by:

1. combining enough tables but not exceeding some acceptable size threshold,
2. selecting only subtables respecting the initial facts,
3. combining the results using the ContextCombine operation and, finally,
4. using the Galois connection to get additional results.

Before making the combination of all the TTBRs associated with the different rules, we propose a preprocessing phase which allows us in many cases to reduce the cost of the construction of the global TTBR. In fact, it is possible to reduce the size of the different TTBRs by starting the reasoning and discovering possibly new facts. All rows with truth value 0 for those new facts are systematically eliminated. After such row eliminations, it is possible to derive new facts using the closure operator. As before, we iteratively reduce the TTBRs as much as possible, i.e., as long as we derive new facts. Only when no more TTBRs can be reduced, we perform the entire combining in order to obtain the global TTBR. Finally, we complete the inference step by selecting additional facts using Galois connection operators on the global TTBR.

**Example 14.** Let us consider the following rules and their corresponding TTBRs in Table 8:

- Rule 1:  $A \rightarrow \neg B \wedge C$
- Rule 2:  $B \rightarrow D$
- Rule 3:  $C \rightarrow E$

A blind initial combination of  $T_1$ ,  $T_2$  and  $T_3$ , without considering initial facts yields a global table with 11 solutions. If instead we assume that we have the initial fact  $A$ , then we can reduce  $T_1$  to the table  $RT_1$  as indicated in Table 9 and



**Table 8**

TTBRs associated with rules 1, 2 and 3, respectively.

A	B	C	B	D	C	E
1	0	1	1	1	1	1
0	0	1	0	1	0	1
0	0	0	0	0	0	0
0	1	1				
0	1	0				
$T_1$			$T_2$		$T_3$	

**Table 9**

Reduced TTBRs.

A	B	C	B	D	C	E
1	0	1	0	1	1	1
			0	0		
$RT_1$			$RT_2$		$RT_3$	

**Table 10**

Combining of three tables.

A	B	C	D	E
1	0	1	0	1
1	0	1	1	1

**Table 11**

Results on the global TTBR.

File name	Time: mm:ss:ms	Global TTBR size
uf20-01.cnf	00:03:046	8 rows
uf20-02.cnf	00:05:390	29 rows
uf20-03.cnf	00:09:953	1 row
uf20-04.cnf	00:02:046	3 rows
uf20-05.cnf	00:23:765	2 rows
uf20-06.cnf	01:38:500	4 rows
uf20-07.cnf	00:02:843	23 rows
uf20-08.cnf	00:20:718	4 rows
uf20-09.cnf	00:02:109	1 rows
uf20-010.cnf	00:06:312	9 rows

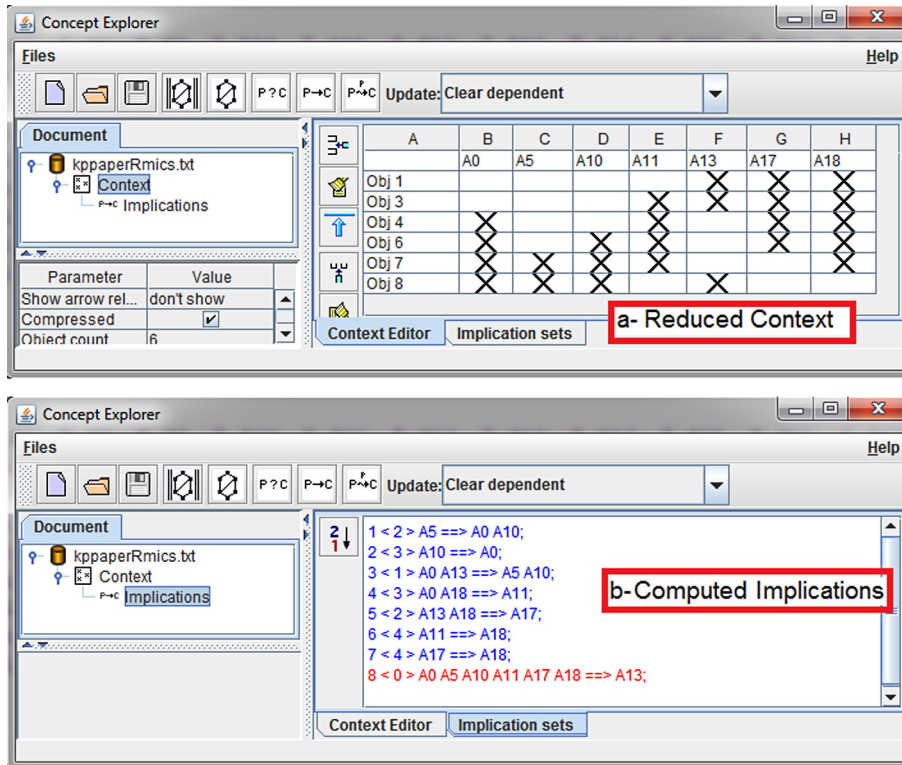
conclude that  $A \rightarrow (\neg B \wedge C)$ . Consequently, we reduce table  $T_2$  to  $RT_2$  (i.e., we remove those rows in  $T_2$  where  $B = 1$ ). Similarly, we reduce table  $T_3$  to  $RT_3$  by removing those rows where  $C = 0$ . Combining now  $RT_1$ ,  $RT_2$  and  $RT_3$  gives the TTBR from Table 10, which only has 2 rows. We could conclude now that  $A \rightarrow (\neg B \wedge C \wedge E)$ .

## 5. Experimental results

In order to study the efficacy of the proposed approach we have considered two case studies. The first one is related to the SAT problem [3], and the second one, provided in [16], relates to Medical Data Rules. In case study 1 we have considered two sub-cases: in the first one (satisfiable sets of rules) we have processed 10 files containing 91 rules and 20 attributes and we have obtained different TTBRs with reduced sizes, which are presented in Table 11. Also, we have presented the TTBR with 8 SAT-solutions for the first file and we have used the ConImp tool in order to derive a reduced set of implications. In the second sub-case (unsatisfiable sets of rules) we have considered 17 UNSAT problems for which the combined TTBRs were, as expected, empty. This result confirms the effectiveness of the approach for inconsistency detection between rules and the possible localization of the sub-set of rules responsible for this inconsistency. In case study 2 we have combined a set of rules related with medical data and we have obtained an improved version of the rules (i.e., the implications extracted from the combined TTBR). There is a concrete application for this: the reduced implications allow us to avoid some redundant medical tests required for diagnosis. The experiment were run in a Laptop Intel(R) Core(TM) i7-3630QM CPU @ 2.40 GHz (8 CPUs), 2.4 GHz, with 16 GB of RAM.

**Table 12**  
TTBR representing the 91 rules.

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
S <sub>1</sub>	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1	0
S <sub>2</sub>	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	1	1	1	0
S <sub>3</sub>	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	0	1	1	1	0
S <sub>4</sub>	1	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	1	1	1	0
S <sub>5</sub>	1	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0
S <sub>6</sub>	1	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	1	1	1	0
S <sub>7</sub>	1	0	0	0	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	0
S <sub>8</sub>	1	1	1	1	0	1	1	1	0	0	1	0	0	1	1	1	1	0	0	1



**Fig. 3.** Using ConImp [12,13] to generate the implications.

5.1. Case study (1): SAT/UnSAT problems

In order to illustrate the inconsistency detection capabilities of this work, as well as the reasoning aspects, we have made some experiments using some satisfiability benchmark problems accessible from <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>.

5.1.1. SAT problem

In Table 11 we summarize the experimental results for 10 files containing satisfiable sets of rules, indicating the requested time for the global combining as well as the size of the obtained TTBRs. As mentioned before, the 10 files have exactly the same size (91 rules and 20 attributes) but the required times for generating the TTBR vary from 2 seconds to less than 2 minutes. At the same time, we remark that the size of the obtained TTBRs is small (from 1 to 29 rows, but most cases have less than 5 rows). This could result in a significant advantage for the reasoning step.

In Table 12 we present the TTBR resulting from processing file uf20-01.cnf, which contains 91 rules and 20 attributes.

Starting from Table 12 we have extracted the set of Duquenne–Guigues implications as indicated in Fig. 3. In the upper screenshot we show the combined TTBR in a reduced format (a reduction option in the tool is used here to avoid redundancy and to eliminate empty columns). The lower screenshot shows the extracted set of implications.

**Table 13**  
Attributes description.

Attribute	Description
trestbps	resting blood pressure (in mm Hg on admission to the hospital)
chol	serum cholesterol in mg/dl
fbs	fasting blood sugar >120 mg/dl
restecg	resting electrocardiogram hic results
thalach	maximum heart rate achieved
Exang	exercise induced angina
oldpeak	ST depression

**Table 14**  
TTBR obtained according to the provided classification rules.

Id	'TMT'	'diagnosis'	'fbs'	'chol'	'restecg'	'trestbps'
1	0	0	0	0	0	0
2	0	0	0	0	0	1
3	0	0	1	0	0	0
4	0	0	1	0	0	1
5	0	1	0	0	0	0
6	0	1	0	0	0	1
7	0	1	0	0	1	0
8	0	1	0	0	1	1
9	0	1	0	1	0	0
10	0	1	0	1	0	1
11	0	1	0	1	1	0
12	0	1	0	1	1	1
13	0	1	1	0	0	0
14	0	1	1	0	0	1
15	0	1	1	0	1	0
16	0	1	1	0	1	1
17	0	1	1	1	0	0
18	0	1	1	1	0	1
19	0	1	1	1	1	0
20	0	1	1	1	1	1
21	1	1	0	0	0	0
22	1	1	0	0	0	1
23	1	1	0	0	1	0
24	1	1	0	0	1	1
25	1	1	0	1	0	0
26	1	1	0	1	0	1
27	1	1	0	1	1	0
28	1	1	0	1	1	1
29	1	1	1	0	0	0
30	1	1	1	0	0	1
31	1	1	1	0	1	0
32	1	1	1	0	1	1
33	1	1	1	1	0	0
34	1	1	1	1	0	1
35	1	1	1	1	1	0
36	1	1	1	1	1	1

### 5.1.2. Inconsistency detection

In Table 15 we present the results obtained for 17 files containing unsatisfiable sets of rules. The first column indicates the file name as well as the number of terms and clauses respectively. For instance, the first file 'aim-50-2\_0-no-4.cnf' (50t 100c) indicates a problem containing 100 clauses (i.e., rules) and 50 terms (i.e., attributes). The second column indicates the time required to combine all TTBRs until an empty one is obtained. The format is hh:mm:ss.ms. Finally, the third column indicates the subset of rules as well as the intermediate ones responsible for the inconsistency (the ones in conflict). Computation of the global TTBR is made progressively, a subset at a time, producing the creation of new intermediate TTBRs. The last ones were identified by incrementing the total rules number. For instance in Table 15, row 1, we have initially 100 rules and 35 intermediate ones. A conflict is detected between rules 130 and 135. The initial subset of rules responsible for the inconsistency could also be located and provided to the final user.

### 5.2. Case study (2) on medical data

In this second case study, we experimented on a set of medical data rules provided in [16], related to a heart diseases dataset available at the UCI site [17]. Each attribute value is obtained by a medical examination test. The considered at-

**Table 15**  
Inconsistency detection.

File name	Time	Inconsistency found (TTBR is empty) between rules
aim-50-2_0-no-4.cnf (50t 100c)	00:08:09:707	[130 135]
aim-100-1_6-no-1.cnf (100t 160c)	00:05:30:929	[137 130 142 143 146 151 154 155 98 111 107 148 147 149 159]
aim-100-1_6-no-2.cnf (100t 160c)	00:06:21:345	[85 86 102 98 99 106 126 125 154 127 120 133 128 108 117]
aim-100-1_6-no-3.cnf (100t 160c)	00:06:45:994	[81 97 86 96 99 91 148 113 122 101 98 105 141 125 112 111 118 114 139 117 107 106 109 144 121 119 120 124 147 129 130 145]
aim-100-1_6-no-4.cnf (100t 160c)	00:06:05:651	[192 197]
aim-100-2_0-no-1.cnf (100t 200c)	00:04:23:339	[116 117 112 141 151 170 118 163 168 108 130 155 156 137 181 158 119 159 115 152 157 164]
aim-100-2_0-no-2.cnf (100t 200c)	00:04:26:489	[242 248 230 239]
aim-100-2_0-no-3.cnf (100t 200c)	00:04:30:413	[209 234 217 218 221 222]
aim-100-2_0-no-4.cnf (100t 200c)	00:07:34:329	[205 238 242 230]
aim-200-1_6-no-1.cnf (200t 320c)	00:05:54:376	[229 297 255 278 203 244 254 294 209 197 256 230 235 187 275 287 273]
aim-200-1_6-no-2.cnf (200t 320c)	00:04:10:881	[187 221 172 132 156 114 181 208 183 186 166 139 101 247 162 111 146 93 155 153 171 199 161 175 182]
aim-200-1_6-no-3.cnf (200t 320c)	00:06:00:062	[248 275 247 242 255 236 195 238 252 230 229 258 271 261 263 274 281]
aim-200-1_6-no-4.cnf (200t 320c)	00:05:36:480	[184 185 141 222 229 217 218 219 176 212 232 214 115 237 223 171 146]
aim-200-2_0-no-1.cnf (200t 400c)	00:06:21:636	[277 278 287 283 275 245 219 215 276 267 280 205 186 292 291 285 305 248 279 386 317]
aim-200-2_0-no-2.cnf (200t 400c)	00:03:34:642	[227 228 238 237 231 148 250 271 325 275 254 270 272 278 279 240 283 399 265 290 299 226]
aim-200-2_0-no-3.cnf (200t 400c)	00:03:04:376	[253 254 228 251 280 236 222 283 272 200 206 285 290 295 292 289]
aim-200-2_0-no-4.cnf (200t 400c)	00:02:58:511	[271 309 345 368 366 365 344 308 353 383 318 340 372 287]

tributes are described in Table 13. Attributes ‘thalach’, ‘Exang’, ‘oldpeak’ and ‘slope’ are the result of performing the medical test ‘TMT’. For our case study we consider the same exact rules presented in [16], namely,

1. if (TMT = ‘+ve’) and (other attributes are ‘+ve’ or ‘ve’) then diagnosis is ‘+ve’.
2. if (TMT = ‘-ve’) and (fbs = ‘+ve’) and (chol = ‘+ve’) and (restecg = ‘+ve’) then diagnosis is ‘+ve’.
3. if (TMT = ‘-ve’) and (fbs = ‘-ve’) and (chol = ‘+ve’) and (restecg = ‘+ve’) then diagnosis is ‘+ve’.
4. if (restecg = ‘+ve’) and (TMT = ‘-ve’) and (chol = ‘+ve’) then diagnosis is ‘+ve’.
5. if (restecg = ‘-ve’) and (TMT = ‘-ve’) and (chol = ‘+ve’) then diagnosis is ‘+ve’.
6. if (trestbps = ‘+ve’) and ((chol = ‘+ve’) or (restecg = ‘+ve’)) and (TMT = ‘-ve’) then diagnosis is ‘+ve’.
7. if (trestbps = ‘-ve’) and ((chol = ‘+ve’) or (restecg = ‘+ve’)) and (TMT = ‘-ve’) then diagnosis is ‘+ve’.

As described in [16] there exists some redundancy within the classification rules. For example, according to classification rules 2 and 3, if TMT gives negative results and if chol and restecg give positive results, then fbs is redundant. Also, according to classification rules 4 and 5, if TMT gives negative results, then if chol gives positive results, then restecg is redundant. Detecting such redundancies helps in deciding the number of medical tests required for diagnosis. The above results can be summarized as [16]:

- Perform TMT test.
- If result is ‘+ve’ then diagnosis is ‘+ve’.
- If result is ‘-ve’ then perform chol test.
- If chol test gives ‘+ve’ as result, then don’t perform trestbps test.

In our case, with Conceptual Reasoning, we have encoded the 7 rules in different TTBRs and have obtained a global TTBR containing 36 solutions as shown in Table 14. Also, we have extracted the following set of implications:

- If TMT result is ‘+ve’ then diagnosis is ‘+ve’.
- If chol test gives ‘+ve’, then diagnosis is ‘+ve’.
- If restecg result is ‘+ve’, then diagnosis is ‘+ve’.

We remark that besides the fact that we were able to determine that there is no inconsistency between the different rules (because the global TTBR is non-empty), the rules we obtained provide important additional information for the restecg test. In fact, a positive result implies ‘+ve’ for diagnosis, which is not mentioned in [16]. This specific case shows to what extent the global conceptual analysis of a set of rules is better than a restrictive analysis on a subset of rules. Furthermore, the global TTBR could be very interesting for experts to possibly infer new relationships between medical tests and diagnoses. Finally, the Cooperative and Conceptual Reasoning alternative provides exactly the same result as the Global Context which is useful in case of a larger rules set.

## 6. Conclusions and further work

We have shown that it is possible to combine truth tables represented as formal contexts, and from these, using a Galois connection, to make forward reasoning without managing conflicts caused by the forward chaining method. We may even reduce the reasoning context (this has already been proved in [18,7]) to speed up the reasoning step, if we use only positive attributes (attributes whose value is 1) as given facts, and seek to obtain goals without negations, which is usual. Cooperative conceptual reasoning is also a solution to speed up the reasoning step. By only calculating partial combining/ of initial tables, upon submission of a subset of facts, the system extracts only those rows satisfying the initial facts. Then, by combining all these sub-tables, we obtain a global table on which we apply the Galois connection to get the conclusions. This method constitutes a new forward reasoning method based on minimal contexts which are equivalent to the initial knowledge base set of rules. It shows efficiency in possible initial conflicts resolutions as well as knowledge expressiveness and reasoning.

## Acknowledgements

This publication was made possible by a grant from the Qatar National Research Fund NPRP 04-1109-1-174. Its contents are solely the responsibility of the authors and do not necessarily reflect the official views of the QNRF. We thank the anonymous reviewers for their valuable comments which raised up the quality of the paper.

## References

- [1] A. Robinson, A. Voronkov, *Handbook of Automated Reasoning*, vols. I & II, Elsevier/MIT Press, 2001.
- [2] S. Owre, J. Rushby, N. Shankar, *A prototype verification system*, in: *Proceedings of the 11th International Conference on Automated Deduction*, in: *Lecture Notes in Artificial Intelligence*, vol. 607, Springer, 1992, pp. 748–752.

- [3] C.P. Gomes, H. Kautz, A. Sabharwal, B. Selman, Satisfiability solvers, in: Frank Van Harmelen, Vladimir Lifschitz, Bruce Porter (Eds.), Handbook of Knowledge Representation, in: Foundations of Artificial Intelligence, vol. 3, Elsevier, 2001.
- [4] J.-L. Laurière, M. Vialatte, Snark: A language to represent declarative knowledge and an inference engine which uses heuristics, in: IFIP Congress, 1986, pp. 811–816.
- [5] K.C. Lee, H.R. Cho, J.-S. Kim, An expert system using an extended and-or graph, *Knowl.-Based Syst.* 21 (2008).
- [6] Z. Wu, G. Eadon, S. Das, E.I. Chong, V. Kolovski, M. Annamalai, J. Srinivasan, Implementing an inference engine for RDFS/OWL constructs and user-defined rules in oracle, in: ICDE 2008, 2008, pp. 1239–1248.
- [7] J.L. Guigues, V. Duquenne, Familles minimales d'implications informatives résultant d'un tableau de données binaires, *Math. Sci. Hum.* 95 (1986) 5–18.
- [8] A. Jaoua, S. Elloumi, Galois connection: Formal concepts and Galois lattice in real relations: Application in a real classifier, *J. Syst. Softw.* 60 (2002) 149–163.
- [9] B. Ganter, R. Wille, *Formal Concept Analysis: Mathematic Foundations*, Springer-Verlag, 1999.
- [10] G. Schmidt, *Relational Mathematics*, Encyclopedia of Mathematics and Its Applications, vol. 132, Cambridge University Press, 2011.
- [11] E. Codd, A relational model of data for large shared data banks, *Commun. ACM* 13 (1970).
- [12] S.A. Yevtushenko, System of data analysis “concept explorer”, in: Proceedings of the 7th National Conference on Artificial Intelligence KII-2000, Russia, 1974, pp. 127–134.
- [13] P. Burmeister, Formal concept analysis with ConImp: Introduction to the basic features, Technical Report, Technische Hochschule, 1998.
- [14] E. Mendelson, *The Propositional Calculus*, 4th ed., Chapman and Hall, 1997, pp. 12–44.
- [15] I. Nafkha, A. Jaoua, Cooperative conceptual retrieval for heterogeneous information, in: ICDIM 2006, 2006.
- [16] A. Gupta, N. Kumar, V. Bhatnagar, Analysis of medical data using data mining and formal concept analysis, *World Acad. Sci., Eng. Technol.* 11 (2007).
- [17] C. Merz, P. Murthy, UCI repository of machine learning database, <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/>, 1996.
- [18] S. Elloumi, J. Jaam, A. Hasnah, A. Jaoua, I. Nafkha, A multi-level conceptual data reduction approach based on the Lukasiewicz implication, *Inf. Sci.* 163 (2004) 253–262.