Applied Mathematical Sciences, Vol. 6, 2012, no. 126, 6253 - 6271

A Formal Procedure for Finding Contradictions into a Set of Rules

Renato Bruni

Universitá di Roma "Sapienza"
Dip. di Ingegneria Informatica, Automatica e Gestionale (DIAG)
Via Ariosto 25, 00185 Roma, Italy
bruni@dis.uniroma1.it

Gianpiero Bianchi

Istat, Dip. per i Censimenti e gli Archivi Amm. e Statistici (DICA)
Viale Oceano Pacifico 171, 00144 Roma, Italy
gianbia@istat.it

Abstract

Several fields of knowledge management operate by using rules. Many example arise in Data Mining and Database Theory, but also in the fields of Normative or Regulation. A major issue is the presence of contradictions into a set of rules, since this usually makes such a set unusable. Each contradiction should therefore be located and removed. We present here an automatic procedure for solving this difficult problem. A main advantage is that this procedure works only at the formal level, so it can be performed without the need of going into the semantic meaning of the rules under analysis. A detailed and realistic example of application of the proposed procedure is given and commented.

Keywords: Alternative Theorems, Inconsistency Selection, Linear Models

1 Introduction

In several fields of knowledge, many tasks are accomplished by using sets of expressions called *rules* (see e.g. [9]). Rules are typically used do detect, among a possibly large set of elements, the ones verifying some condition. This happens for example in Data Mining, in Database Theory, in Statistics, but also in less mathematical fields such like Normative or Regulation. The condition may be of any nature, for instance "being correct", "being wrong", "being convenient", "respecting the laws", "being compliant with a standard", etc. The set of rules may have several origins: it could be automatically generated, for instance learned by some dataset, or be written by human experts, or also be the result of an updating or a merging of other sets of rules. A major issue is the presence of contradictions into the set of rules itself. This can frequently arise, in particular when the set of rules has been assembled from different sources. Generally, the presence of contradictions makes such a set not usable anymore. Each contradiction should therefore be located and removed, either by deleting or by slightly changing some of the rules. This is however a very difficult problem in general: a contradiction can be quite hidden, or involve many rules, or there can be several contradictions. Moreover, this difficulty rapidly increases with the size of the set of rules [10].

We present here an automatic procedure for finding a contradiction into a set of rules. The procedure can be iterated until all contradictions are removed from a set. A main advantage of the proposed approach is that this procedure works only at the formal level, so it can be performed without the need of going into the semantic meaning of the rules under analysis and can be applied to rules arising from any field. In particular, Section 2 explains how several kind of rules can be formally represented into linear inequalities. After this, Section 3 presents a theoretical condition, based on a variant of Farkas' lemma (see e.g. [14]), used to detect a single contradiction. All contradictions are detected by iterating this procedure, and the structure of the set of all contradictions, together with the relationships among themselves, are also studied. Finally, Section 4 gives a detailed explanation of the operations performed by the proposed procedure on a realistic set of rules.

2 Encoding Rules into Linear Inequalities

In Database theory, a *record schema* is a set of fields f_i, with $i = 1 \dots m$, and a *record instance* is a set of values v_i, one for each of the above fields. In order to help exposition, we will focus on records representing *persons*. Note, however, that the proposed procedure is not influenced by the meaning of processed data. The record scheme will be denoted by P, whereas a generic record instance corresponding to P will be denoted by p.

$$P = \{f_1, \dots, f_m\} \qquad p = \{v_1, \dots, v_m\}$$

Example 2.1. For records representing persons, fields are for instance `age` or `marital status`, and corresponding examples of values are `18` or `single`.

Each field f_i, with $i = 1 \dots m$, has its *domain* D_i, which is the set of every possible value for that field. A distinction is usually made between *quantita-*

tive, or *numerical*, fields, and *qualitative*, or *categorical* fields. The proposed approach is able to deal with both qualitative and quantitative values.

In several applications, records verifying some condition are selected by using *rules*. Each rule can be seen as a mathematical function r_k from the Cartesian product of all the domains to the Boolean set {0,1}, as follows.

$$\begin{array}{rccc} r_k: & D_1 \times \ldots \times D_m & \rightarrow & \{0,1\} \\ & p & \mapsto & 0,1 \end{array}$$

We call *logical rules* the rules expressed only with logical conditions, *mathematical rules* the rules expressed only with mathematical conditions, and *logic-mathematical rules* the rules expressed using both types of condition. See [3] for further details on different kind of rules.

Values appearing in the rules are called *breakpoints*, or *cut points*, for the domains. They represent the logical *watershed* between values of the domain, and will be indicated with b_i^j. Such breakpoints are used to split every domain D_i into n_i subsets S_i^j representing values of the domain which are *equivalent* from the rules' point of view. We congruently have $D_i = \bigcup_{j=1}^{n_i} S_i^j$.

Example 2.2. Suppose that, by scanning a given set of rules R, the following breakpoints are obtained for the field `age` of a person.

$$b^1_{\texttt{age}} = \texttt{0},\ b^2_{\texttt{age}} = \texttt{14},\ b^3_{\texttt{age}} = \texttt{18},\ b^4_{\texttt{age}} = \texttt{26},\ b^5_{\texttt{age}} = \texttt{110},\ b^6_{\texttt{age}} = \texttt{blank}$$

and, by using the breakpoints and the rules to cut $D_{\texttt{age}}$, we have the $n_{\texttt{age}} = 5$ subsets. The last subset is the out-of-range one.

$$\begin{gathered} S_{\texttt{age}\in\{0\ldots13\}} = \{\texttt{0},\ldots,\texttt{13}\},\ S_{\texttt{age}\in\{14\ldots17\}} = \{\texttt{14},\ldots,\texttt{17}\}, \\ S_{\texttt{age}\in\{18\ldots25\}} = \{\texttt{18},\ldots,\texttt{25}\},\ S_{\texttt{age}\in\{26\ldots110\}} = \{\texttt{26},\ldots,\texttt{110}\}, \\ S_{\texttt{age = out\_of\_range}} = \{\ldots,-\texttt{1}\} \cup \{\texttt{111},\ldots\} \cup \{\texttt{blank}\} \end{gathered}$$

Now, the *variables* for the announced linear inequalities can be introduced: a set of m real variables $z_i \in [0, U]$, one for each domain D_i, and a set of $n = n_1 + \ldots + n_m$ binary variables $x_{ij} \in \{0,1\}$, one for each subset S_{ij}. We represent each value v_i of p with a real variable z_i, by defining a mapping φ between values of the domain and real numbers between 0 and an upper value U. Note that, occasionally, it could be convenient to bound some of the z_i variables to be integer, as described in [3], with obvious specific modifications in the rest of the procedure. However, we continue our description considering the general case of real z variables.

The membership of a value v_i to the subset S_{ij} is encoded by using the binary variables x_{ij}.

$$x_{ij} = \begin{cases} 1 & \text{when} \quad v_i \in S_{ij} \\ 0 & \text{when} \quad v_i \notin S_{ij} \end{cases}$$

Binary and real variables are linked by using a set of linear inequalities called *bridge constraints*. They impose that, when z_i has a value such that v_i belongs to subset S_{ij}, the corresponding x_{ij} is 1 and all others binary variables $\{x_{i1}, \ldots, x_{ij-1}, x_{ij+1}, \ldots, x_{in_i}\}$ of field f_i are 0. By using these variables, all the above types of rule can be expressed. For further details see [3, 4].

Example 2.3. Consider the following logical rule.

$$\neg(\texttt{marital status} = \texttt{married}) \vee \neg(\texttt{age} < 14)$$

By substituting the logical conditions, it becomes the linear inequality:

$$(1 - x_{\texttt{marital\_status = married}}) + (1 - x_{\texttt{age}\in\{0\ldots13\}}) \geq 1$$

Consider, instead, the following logic-mathematical rule.

$$\neg(\texttt{marital status} \ = \ \texttt{married}) \vee (\texttt{age} - \texttt{years married} \geq 14)$$

By substituting the logical and mathematical conditions, we have

$$(1 - x_{\texttt{marital status = married}}) \vee (z_{\texttt{age}} - z_{\texttt{years married}} \geq 14)$$

which becomes the following linear inequality

$$U(1 - x_{\texttt{marital status = married}}) + z_{\texttt{age}} - z_{\texttt{years married}} \geq 14$$

Altogether, from the set of rules R, a set of linear inequalities is obtained. Each record p determines an assignment of values for the introduced variables x_{ij} and z_i. By denoting with x and z the vectors respectively made of all the components x_{ij} and z_i, $i = 1 \ldots m$, $j = 1 \ldots n_i$, as follows,

$$x = (x_{11}, \ldots, x_{1n_1}, \ldots, x_{m1}, \ldots, x_{mn_m})^T \quad z = (z_1, \ldots, z_m)^T$$

the set of rules R becomes a system of linear inequalities, expressed in compact notation as follows.

$$\begin{cases} B \begin{bmatrix} x \\ z \end{bmatrix} \geq b \\ 0 \leq z_i \leq U \qquad i = 1 \ldots m \\ x \ \in \ \{0,1\}^n \\ z \ \in \ \mathbb{R}^m \end{cases} \tag{1}$$

Since x has $n = n_1 + ... + n_m$ components and z has m components, and letting l be the total number of inequalities, B is in general a $l \times (n+m)$ real matrix, and b a real l-vector.

3 Locating Contradictions

A contradiction in the set of rules corresponds to an unsatisfiable set of inequalities within the above described system of linear inequalities. Such an unsatisfiable set is called *Infeasible Subsystem* (IS). When an IS is minimal, i.e. becomes satisfiable by removing anyone of its inequalities, is called *Irreducible Infeasible Subsystem*(IIS) [1, 7, 15]. In the case of systems of linear inequalities having real variables, the problem has been approached both by means of heuristics [6] and exact algorithms [11]. In the case of systems of linear inequalities having integer variables (more computationally demanding), the problem has been approached by means of additive or subtractive heuristics [12]. We propose here a procedure based on a variant of well known Farkas' lemma adapted from the continuous to the discrete case.

Theorem 3.1 (Farkas' lemma) *Let A be an $s \times t$ real matrix and let a be a real s-vector. Then there exists a real t-vector $x \geq \mathbf{0}$ with $Ax = a$ if and only if $y^T a \geq 0$ for each real s-vector y with $y^T A \geq \mathbf{0}$.*

Geometrically, this means that if an s-vector γ does not belong to the cone generated by the s-vectors $a_1, \ldots, a_t$ (columns of A), there exists a linear hyperplane separating γ from $a_1, \ldots, a_t$. There are several equivalent forms of Farkas' lemma. The following variant is more suitable to our purposes. Given a matrix $A \in \mathbb{R}^{s \times t}$ and a vector $a \in \mathbb{R}^s$, consider the system:

$$\left\{ \begin{array}{rcl} Ax & \leq & a \\ x & \in & \mathbb{R}^t \end{array} \right. \tag{2}$$

and the new system of linear inequalities obtained from the former one:

$$\left\{ \begin{array}{rcl} y^T A & = & \mathbf{0} \\ y^T a & < & 0 \\ y & \geq & \mathbf{0} \\ y & \in & \mathbb{R}^s \end{array} \right. \tag{3}$$

We have that exactly one of the two following possibilities holds:

- (2) is feasible, i.e. there exists $x \in \mathbb{R}^t$ verifying all its inequalities.
- (3) is feasible, i.e. there exists $y \in \mathbb{R}^s$ verifying all its inequalities.

An IIS can be selected within (2) by solving the following new system [11]:

$$\left\{ \begin{array}{rcl} y^T A & = & \mathbf{0} \\ y^T a & \leq & -1 \\ y & \geq & \mathbf{0} \\ y & \in & \mathbb{R}^s \end{array} \right. \tag{4}$$

The *support* of a vertex denotes the indices of its non-zero components; **0**, **1** and **U** respectively denote vectors of zeroes, ones and Us of appropriate dimension.

Theorem 3.2. (Gleeson and Ryan) *Consider two systems of linear inequalities respectively in form (2) and (4). If (4) is infeasible, (2) is feasible. On the contrary, if (4) is feasible, (2) is infeasible, and, moreover, each IIS of (2) is given by the support of each vertex of the polyhedron (4).*

The proof is based on polyhedral arguments using properties of extreme rays, see [11]. Therefore, checking the feasibility of (2), and, if infeasible, identifying one of its IIS, becomes the problem of finding a vertex of a polyhedron, that can be easily solved (e.g. with the simplex algorithm [2, 14]).

However, in the case of (1), we have a systems of linear inequalities were we are interested in mixed-integer solutions. In order to use the results given for the linear case, let us consider the linear relaxation of such system (1).

$$\left\{ \begin{array}{lcl} -B\begin{bmatrix} x \\ z \end{bmatrix} & \leq & -b \\ \begin{bmatrix} x \\ z \end{bmatrix} & \leq & \begin{bmatrix} \mathbf{1} \\ \mathbf{U} \end{bmatrix} \\ -\begin{bmatrix} x \\ z \end{bmatrix} & \leq & \mathbf{0} \\ \begin{bmatrix} x \\ z \end{bmatrix} & \in & \mathbb{R}^{n+m} \end{array} \right. \tag{5}$$

The above system (5) is now in the form of (2). The l inequalities from the first group will be called *rules inequalities*, even if, for some of them, there can be no one-to-one correspondence with rules (see Sect. 4). By denoting with I the identity matrix, the $[l + 2(n + m)] \times (n + m)$ matrix A and the $[l+2(n+m)]$-vector a are composed as follows. Number of rows for each block is reported on the left.

$$A = \begin{bmatrix} -B \\ I \\ -I \end{bmatrix} \begin{array}{c} l \\ n+m \\ n+m \end{array} \qquad a = \begin{bmatrix} -b \\ \mathbf{1} \\ \mathbf{U} \\ \mathbf{0} \end{bmatrix} \begin{array}{c} l \\ n \\ m \\ n+m \end{array}$$

Therefore, a system which plays the role of (4) can now be written.

$$\begin{cases} y^T \begin{bmatrix} -B \\ I \\ -I \end{bmatrix} & = \ \mathbf{0} \\ y^T \begin{bmatrix} -b \\ \mathbf{1} \\ \mathbf{U} \\ \mathbf{0} \end{bmatrix} & \leq \ -1 \\ y \ \geq \ \mathbf{0}, \quad y \in \ \mathbb{R}^{[l+2(n+m)]} \end{cases} \tag{6}$$

So far, the following result on the pair of systems (1) and (6) holds. The *restriction* of the support of a vertex to rules inequalities will denote the indices of its non-zero components among those corresponding to rules inequalities.

Theorem 3.3. *Consider two systems of linear inequalities respectively in form (1) and (6). In this case, if (6) is feasible, (1) is infeasible, and the restriction of the support of each vertex of the polyhedron (6) to rules inequalities contains an IIS of (1). On the contrary, if (6) is infeasible, (5) is feasible, but it cannot be decided whether (1) is feasible or not.*

Proof: We first prove that the restriction of the support of a vertex of (6) to rule inequalities contains an integer IIS of (1). Assume (6) is feasible, and let v_1 be the vertex found. Therefore, (5) is infeasible (from Theorem 3.1), and an IIS in (5), called here IIS_1, is given by the support of v_1. Such IIS_1 is in general composed by a set RI_1 of *rules inequalities* and a set BC_1 (possibly empty) of *box constraints* (the ones imposing $0 \leq x_{ij} \leq 1, 0 \leq z_i \leq U$). The set of inequalities RI_1 has no integer solutions, since removing the BC_1 from IIS_1, while imposing the more strict *integer constraints* IC_1 (the ones imposing $x_{ij} \in \{0,1\}$), keeps IIS_1 unsatisfiable. Therefore, an integer IIS is contained into RI_1. The integer IIS may also be a subset of the inequalities of RI_1, because, though $IIS_1 = RI_1 \cup BC_1$ is minimally infeasible, $RI_1 \cup IC_1$ may be not minimal: we are imposing the more strict integer constraints instead of the box constraints. Therefore, the procedure produces an integrally infeasible subsystem containing an integer IIS for (1).

On the other hand, not all integer IIS in (2) can be obtained by such procedure. This because, if (6) is infeasible, (5) is feasible (by Theorem 3.1). When imposing the more strict integer constraints instead of the box constraints, however, nothing can be said on the feasibility of (1).

Example 3.1. Consider a set of rules R on two conditions α_1, α_2, as follows. One may already note that R contains an inconsistency.

$$r_1 = (\alpha_1),\ r_2 = (\alpha_2),\ r_3 = (\neg\alpha_1 \vee \neg\alpha_2),\ r_4 = (\alpha_1 \vee \neg\alpha_2)$$

In this case, $n = 2$ and m can be considered 0, since no z variables are needed to express the above rules. A and a can easily be obtained, as follows.

$$A = \left[\begin{array}{rr} -1 & 0 \\ 0 & -1 \\ 1 & 1 \\ -1 & 1 \\ \hline 1 & 0 \\ 0 & 1 \\ \hline -1 & 0 \\ 0 & -1 \end{array}\right] \qquad a = \left[\begin{array}{r} -1 \\ -1 \\ 1 \\ 0 \\ \hline 1 \\ 1 \\ \hline 0 \\ 0 \end{array}\right]$$

Therefore, the system to be solved, in the form of (6), is the following.

$$\left\{\begin{array}{rcl} -y_1 + y_3 - y_4 + y_5 - y_7 & = & 0 \\ -y_2 + y_3 + y_4 + y_6 - y_8 & = & 0 \\ -y_1 - y_2 + y_3 + y_5 + y_6 & \leq & -1 \\ y_1,\, y_2,\, y_3,\, y_4,\, y_5,\, y_6,\, y_7,\, y_8 & \geq & 0 \\ y & \in & \mathbb{R}^8 \end{array}\right.$$

Solving such system yields the vertex (1, 1, 1, 0, 0, 0, 0, 0). Therefore, R contains an inconsistency, and the set of conflicting rules is $\{r_1, r_2, r_3\}$.

More than one IIS can be contained in an infeasible system. Some of them can overlap, in the sense that they can share some inequalities, although they cannot be fully contained one in another. Formally, the collection of all IIS of a given infeasible system is a *clutter* (see e.g. [1]). However, from the practical point of view, we are interested in IIS composed by a small number of rules inequalities. Moreover, it may happen that not all of them are equally preferable for the composition of the IIS that we are selecting. Hence, a cost c_k for taking each of the $[l+2(n+m)]$ inequalities into our IIS can be assigned. Such costs c_k for the inequalities of (5) corresponds to costs for the variables of system (6). A cost $[l + 2(n + m)]$-vector c is therefore computed, and the solution of the following linear program produces now an IIS having the desired inequality composition.

$$\left\{\begin{array}{l} \min\ c^T y \\ y^T \begin{bmatrix} -B \\ I \\ -I \end{bmatrix} = \mathbf{0} \\ y^T \begin{bmatrix} -b \\ \mathbf{1} \\ \mathbf{U} \\ \mathbf{0} \end{bmatrix} \leq -1 \\ y \geq \mathbf{0}, \quad y \in \mathbb{R}^{[l+2(n+m)]} \end{array}\right. \qquad (7)$$

The result of Theorem 3.3 is not completely analogous to the linear case. In order to obtain more analogy, let us define the following property.

Integral-point property. A class of polyhedra which, if non-empty, contain at least one integral point (i.e. a point respecting integrality constraints) has the integral-point (IP) property.

Theorem 3.4. *If the polyhedron (5), which is the linear relaxation of (1), has the integral-point property, the following holds. If (6) is infeasible, (1) is feasible. On the contrary, if (6) is feasible, (1) is infeasible and each integer IIS is given by the restriction of the support of each vertex of polyhedron (6) to rules inequalities.*

Proof: If (6) is infeasible, (5) is feasible by Theorem 3.1. Since we assumed that the IP-property holds for (5), it contains at least one integral point. Since the box constraints hold for (5), this integer point must be such that $x \in \{0,1\}^n$, hence (1) is feasible. On the contrary, if (6) is feasible, the restriction of the support of a vertex in (6) to rule inequalities, that is a set of inequalities denoted by RI_1, has no integer solutions by Theorem 3.3. We now prove by contradiction that RI_1 is minimally infeasible, hence it is an integer IIS. Suppose RI_1 not minimal; then there exists a smaller set RI_1' such that $RI_1' \cup IC_1$ has no integer solutions. On the other hand, by Theorem 3.2, $RI_1 \cup BC_1$ is minimal, so $RI_1' \cup BC_1$ must be feasible, and since it has the IP-property, it has an integer solution, which is the contradiction. The thesis follows.

So far, when the IP property holds, solving a linear programming problem solves our inconsistency selection problem. There are several cases in which the linear relaxation (5) defines a polyhedron having the integral-point property (see e.g. [5, 8, 13]). Note that, imposing some syntactic restrictions, rules could be written in order to obtain one of such cases.

4 Applying the Proposed Procedure

Assume that each individual is described by a data record (a set of values for a set of fields). Let the fields be either categorical, e.g. *name, profession, tax1* (= if the individual has to pay a tax called tax1), *tax2*, *tax3*, or numerical, e.g. *age, length_of_career, income.*

Let the domain of *profession* be a set of strings (e.g. pr1, pr2, pr3); *blank* being an admissible value, e.g. for non-working people); the domain of *tax1, tax2, tax3* be $\{yes, no\}$; the domain of *age* be a suitable subset of the set of real non-negative numbers $\mathbb{R}_+$ (or of Z_+, with obvious modifications); the domain of *length_of_career* be a suitable subset of $\mathbb{R}_+ \cup blank$ (*blank* being an admissible value, e.g. for non-working people); the domain of *income* be a suitable subset of R_+ (being 0 for non-working people).

Assume there is a set of rules for economical regulation (something similar to laws), as follows. Clearly, the focus is not on numerical values appearing in the rules, that may be unrealistic, but on the structure of the set. Note that, in order to test the consistency of this set, we need to consider also rules that a human would consider obvious, but not a machine, called *unexpressed* rules.

- *Logical* rules

  Some taxes must be paid for some professions

L1 if *profession* = (*pr1* or *pr2*) then *tax1* must be *yes*

L2 if *profession* = *pr3* then *tax2* must be *yes*

Some taxes must be paid for some income values

L3 if *income* $\geq$ 1000 then *tax3* must be *yes*

For poor people taxes cannot exceed 100

L4 if *income* $\leq$ 200 then *total_tax* must be $\leq$ 100

- *Mathematical* rules

  Income must be related to length of career

M1 $income \leq 1000 + 20 \times length\_of\_career$

M2 $income \geq 200 + 30 \times length\_of\_career$

Taxes must be at least one third of the income

M3 $total\_tax \geq 0.33 \times income$

Taxes cannot exceed income

M4 $total\_tax \leq income$

- *Logico-mathematical* rules

  If income is too high for the career, tax 3 must me paid

LM1 if $income - 30 \times length\_of\_career \geq 400$ then *tax3* must be *yes*

- *Unexpressed* rules

  Professions are mutually exclusive

U1 $Pr1 \oplus Pr2 \oplus Pr3$

There are relations implied by the meaning of the words

U2 $total\_tax = tax1 + tax2 + tax3$

Some Fields are naturally limited

U3 *age* ≥ 0 and ≤ 110

U4 *length_of_career* ≥ 0 and ≤ 92

U5 $\varepsilon \geq 0$ and ≤ 0.001

U6 *total_tax* ≥ 0 and ≤ 2000

U7 *income* ≥ 0 and ≤ 5000

From the above rules we can identify some *variables*. Some of them are logical, and are also called propositions, and some are real-valued.

1) XPRO1 (binary)
2) XPRO2 (binary)
3) XPRO3 (binary)
4) XTAX1 (binary)
5) XTAX2 (binary)
6) XTAX3 (binary)
7) XTTAX0-100 (binary)
8) XINC0-200 (binary)
9) TTAX (real≥ 0)
10) INC (real≥ 0)
11) AGE (real≥ 0)
12) LEN (real≥ 0)
13) EPS (real≥ 0)

In the general case, from the rules we can identify some logical propositions, that are the elementary concepts expressed in the rules. We may have:

- *Level* propositions, e.g. $L1, L2, L3, L4$. They are conditions that become stronger as their index increases, so $L4 \Rightarrow L3, L2, L1$ and $L3 \Rightarrow L2, L1$ and $L2 \Rightarrow L1$ and $L1$ does not imply anything. Conversely, $\neg L1 \Rightarrow \neg L2, \neg L3, \neg L4$ and $\neg L2 \Rightarrow \neg L3, \neg L4$ and $\neg L3 \Rightarrow \neg L4$ and $\neg L4$ does not imply anything. A set of level propositions is *complete* when at least one of them must hold, so $L1$ is always true.

  They can represent for instance that the value of a certain field of some data records belongs to some sets $S1, S2, S3, S4$ in a domain S such that $S1 \supseteq S2 \supseteq S3 \supseteq S4$ (and are complete when $S1 = S$).

- *Exclusive* propositions, e.g. $E1, E2, E3$. They are mutually exclusive: at most one of them holds, so $E1 \Rightarrow \neg E2, \neg E3$ and $E2 \Rightarrow \neg E1, \neg E3$ and $E3 \Rightarrow \neg E1, \neg E2$. Equivalently, $\neg E1 \vee \neg E2$ and $\neg E2 \vee \neg E3$ and $\neg E1 \vee \neg E3$. A set of exclusive propositions is *complete* when at least one of them must hold, so $E1 \vee E2 \vee E3$.

  They can represent for instance that the value of a certain field of some data records belongs to some sets $S1, S2, S3$ such that $S1 \cap S2 = \phi$ and $S2 \cap S3 = \phi$ and $S1 \cap S3 = \phi$ (complete when $S1 \cup S2 \cup S3 = S$).

- *Standard* propositions, e.g. F, G, H, I. They have no predefined relations among them, and any relation among them can be expressed, e.g. $F \Rightarrow G$ and $F \wedge H \Rightarrow I$.

The rules may contain one or more inconsistency, as explained in Section 3. Note that inconsistencies may be either *complete*, when no record can respect the rules, or *partial*, when no record having a specific value v_i for a specific field i (value that should not be forbidden) can respect the rules. In this example we have:

- No complete inconsistency: there are records respecting all the rules.
- A partial inconsistency for *length_of_career* ≥ 67
(M2 says *income* ≥ 2210 and M3 says *total_tax* ≥ 729.3, while U2 says *total_tax* can be at most 720 when all *tax1, tax2, tax3* are paid. Since U7 says $0 \leq$ *income* ≤ 5000, that is a contradiction).
- Another partial inconsistency for *length_of_career* ≥ 81
(M1 says *income* ≤ 2620 while M2 says *income* ≥ 2630, that is a contradiction).
- Another partial inconsistency for *income* ≥ 2182
(M3 says *total_tax* ≥ 720.06, while U2 says *total_tax* can be at most 720 when all *tax1, tax2, tax3* are paid. Since U7 says $0 \leq$ *income* ≤ 5000, that is a contradiction).

Partial inconsistencies can be tested with the proposed procedure by simply imposing the value activating them, for instance by adding a constraint. We now analyze the above three examples with our procedure. First we convert rules into inequalities, until putting all of them in the form $\leq$

L1 if *profession* = (*pr1* or *pr2*) then *tax1* must be *yes*

= *profession* = ¬ XPRO1 ∨ XTAX1 and ¬ XPRO1 ∨ XTAX1

= XTAX1 + (1-XPRO1) ≥ 1 and XTAX1 + (1-XPRO2) ≥ 1

1) -1 XTAX1 +1 XPRO1 ≤ 0

2) -1 XTAX1 +1 XPRO2 ≤ 0

L2 if *profession* = *pr3* then *tax2* must be *yes*

= XTAX2 + (1-XPRO3) ≥ 1

3) -1 XTAX2 +1 XPRO3 ≤ 0

L3 if *income* ≥ 1000 then *tax3* must be *yes*

= $\neg tax3 \Rightarrow$ *income* < 1000

= *tax3* ∨ *income* $\leq 1000 - \varepsilon$

= -M TAX3 +INC +EPS $\leq$ 1000

4) -M TAX3 +1 INC +1 EPS $\leq$ 1000

L4 if *income* ≤ 200 then *total_tax* must be ≤ 100

= (1-XINC0-200) + XTTAX0-100 ≥ 1

5) 1 XINC0-200 -1 XTTAX0-100 ≤ 0

M1 *income* $\leq 1000 + 20\times$*length_of_career*

= INC -20 LEN $\leq$ 1000

6) 1 INC -20 LEN $\leq$ 1000

M2 *income* $\geq 200 + 30\times$*length_of_career*

= INC -30 LEN $\geq$ 200

7) -1 INC +30 LEN ≤ -200

M3 *total_tax* $\geq 0.33\times$*income*

= TTAX -0.33 INC ≥ 0

8) -1 TTAX +0.33 INC ≤ 0

M4 *total_tax* $\leq$*income*

= TTAX -INC ≤ 0

9) 1 TTAX -1 INC ≤ 0

LM1 if *income* $-30\times$*length_of_career* ≥ 400 then *tax3* must be *yes*

= -M TAX3 + 30 LEN -INC +EPS $\leq$ 400

10) -M TAX3 +30 LEN -1 INC +1 EPS $\leq$ 400

U1 *Pr1* $\oplus$*Pr2* $\oplus$*Pr3*

= PRO1 +PRO2 ≤ 1 and PRO1 +PRO3 ≤ 1 and PRO2 +PRO3 ≤ 1

11) 1 PRO1 +1 PRO2 ≤ 1

12) 1 PRO1 +1 PRO3 ≤ 1

13) 1 PRO2 +1 PRO3 ≤ 1

U2 *total_tax = tax1+tax2+tax3* (with tax1=100, tax2=120, tax3=500)

= TTAX -100 XTAX1 -120 XTAX2 -500 XTAX3 = 0

14) 1 TTAX -100 XTAX1 -120 XTAX2 -500 XTAX3 ≤ 0

15) -1 TTAX +100 XTAX1 +120 XTAX2 +500 XTAX3 ≤ 0

- XTTAX0-100=1 iff TTAX$\leq$ 100

= M (1-XTTAX0-100) $\geq$ TTAX -100 and -M XTTAX0-100 $\geq$ TTAX -100 -EPS

16) M XTTAX0-100 +1 TTAX $\leq$ M+100

17) M XTTAX0-100 +1 TTAX -1 EPS$\leq$ 100

- XINC0-200=1 iff INC$\leq$ 200

= M (1-XINC0-200) $\geq$ INC -200 and M XINC0-200 $\geq$ 200 +EPS - INC

18) M XINC0-200 +1 INC $\leq$ M +200

19) -M XINC0-200 -1 INC +1 EPS $\leq$ -200

U3 *age* ≥ 0 and ≤ 110

= AGE ≥ 0 and AGE ≤ 110

20) -1 AGE ≤ 0 21) 1 AGE ≤ 110

U4 *length_of_career* ≥ 0 and ≤ 92

= LEN ≥ 0 and LEN ≤ 92

22) -1 LEN ≤ 0 23) 1 LEN ≤ 92

U5 $\varepsilon \geq 0$ and ≤ 0.001

= EPS ≥ 0 and EPS ≤ 0.001

24) -1 EPS ≤ 0 25) 1 EPS ≤ 0.001

U6 *total_tax* ≥ 0 and ≤ 2000

= TTAX ≥ 0 and ≤ 2000

26) -1 TTAX ≤ 0 27) 1 TTAX ≤ 2000

U7 *income* ≥ 0 and ≤ 5000

= INC ≥ 0 and INC ≤ 5000

28) -1 INC ≤ 0 29) 1 INC ≤ 5000

- XPRO1 binary

30) -1 XPRO1 ≤ 0 31) 1 XPRO1 ≤ 1

- XPRO2 (binary)

32) -1 XPRO2 ≤ 0 33) 1 XPRO2 ≤ 1

- XPRO3 (binary)

34) -1 XPRO3 ≤ 0 35) 1 XPRO3 ≤ 1

- XTAX1 (binary)

36) -1 XTAX1 ≤ 0 37) 1 XTAX1 ≤ 1

- XTAX2 (binary)

38) -1 XTAX2 ≤ 0 39) 1 XTAX2 ≤ 1

- XTAX3 (binary)

40) -1 XTAX3 ≤ 0 41) 1 XTAX3 ≤ 1

- XTTAX0-100 (binary)

42) -1 XTTAX0-100 ≤ 0 43) 1 XTTAX0-100 ≤ 1

- XINC0-200 (binary)

44) -1 XINC0-200 ≤ 0 45) 1 XINC0-200 ≤ 1

Overall, we have the following set of linear inequalities in form $\leq$

1) -1 XTAX1 +1 XPRO1 ≤ 0
2) -1 XTAX1 +1 XPRO2 ≤ 0
3) -1 XTAX2 +1 XPRO3 ≤ 0
4) -M TAX3 +1 INC +1 EPS ≤ 1000
5) 1 XINC0-200 -1 XTTAX0-100 ≤ 0
6) 1 INC -20 LEN ≤ 1000
7) -1 INC +30 LEN ≤ -200
8) -1 TTAX +0.33 INC ≤ 0
9) 1 TTAX -1 INC ≤ 0
10) -M TAX3 +30 LEN -1 INC +1 EPS ≤ 400
11) 1 PRO1 +1 PRO2 ≤ 1
12) 1 PRO1 +1 PRO3 ≤ 1
13) 1 PRO2 +1 PRO3 ≤ 1
14) 1 TTAX -100 XTAX1 -120 XTAX2 -500 XTAX3 ≤ 0
15) -1 TTAX +100 XTAX1 +120 XTAX2 +500 XTAX3 ≤ 0
16) M XTTAX0-100 +1 TTAX $\leq$ M+100
17) M XTTAX0-100 +1 TTAX -1 EPS$\leq$ 100
18) M XINC0-200 +1 INC $\leq$ M +200
19) M XINC0-200 +1 INC -1 EPS ≤ 200

20) -1 AGE ≤ 0	21) 1 AGE ≤ 110
22) -1 LEN ≤ 0	23) 1 LEN ≤ 92
24) -1 EPS ≤ 0	25) 1 EPS ≤ 0.001
26) -1 TTAX ≤ 0	27) 1 TTAX ≤ 2000
28) -1 INC ≤ 0	29) 1 INC ≤ 5000
30) -1 XPRO1 ≤ 0	31) 1 XPRO1 ≤ 1
32) -1 XPRO2 ≤ 0	33) 1 XPRO2 ≤ 1
34) -1 XPRO3 ≤ 0	35) 1 XPRO3 ≤ 1
36) -1 XTAX1 ≤ 0	37) 1 XTAX1 ≤ 1
38) -1 XTAX2 ≤ 0	39) 1 XTAX2 ≤ 1
40) -1 XTAX3 ≤ 0	41) 1 XTAX3 ≤ 1
42) -1 XTTAX0-100 ≤ 0	43) 1 XTTAX0-100 ≤ 1
44) -1 XINC0-200 ≤ 0	45) 1 XINC0-200 ≤ 1

By ordering the binary (x) and real variables (z), the overall matrix and the overall vector of system (5), corresponding to A and a of system (2), unless an easy reordering of some of the box inequalities, are the following:

	XPRO 1	XPRO 2	XPRO 3	XTAX 1	XTAX 2	XTAX 3	XTTAX 0-100	XINC 0-200	TTAX	INC	AGE	LEN	EPS	vector a
1 (L1)	1	0	0	-1	0	0	0	0	0	0	0	0	0	0
2 (L1)	0	1	0	-1	0	0	0	0	0	0	0	0	0	0
3 (L2)	0	0	1	0	-1	0	0	0	0	0	0	0	0	0
4 (L3)	0	0	0	0	0	-12000	0	0	0	1	0	0	1	1000
5 (L4)	0	0	0	0	0	0	-1	1	0	0	0	0	0	0
6 (M1)	0	0	0	0	0	0	0	0	0	1	0	-20	0	1000
7 (M21)	0	0	0	0	0	0	0	0	0	-1	0	30	0	-200
8 (M3)	0	0	0	0	0	0	0	0	-1	0.33	0	0	0	0
9 (M4)	0	0	0	0	0	0	0	0	1	-1	0	0	0	0
10 (LM1)	0	0	0	0	0	-12000	0	0	0	-1	0	30	1	400
11 (U1)	1	1	0	0	0	0	0	0	0	0	0	0	0	1
12 (U1)	1	0	1	0	0	0	0	0	0	0	0	0	0	1
13 (U1)	0	1	1	0	0	0	0	0	0	0	0	0	0	1
14 (U2)	0	0	0	-100	-120	-500	0	0	1	0	0	0	0	0
15 (U2)	0	0	0	100	120	500	0	0	-1	0	0	0	0	0
16 (U2)	0	0	0	0	0	0	12000	0	1	0	0	0	0	12100
17 (U2)	0	0	0	0	0	0	-12000	0	-1	0	0	0	1	-100
18 (U2)	0	0	0	0	0	0	0	12000	0	1	0	0	0	12200
19 (U2)	0	0	0	0	0	0	0	-12000	0	-1	0	0	1	-200
20 (U3)	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
21 (U3)	0	0	0	0	0	0	0	0	0	0	1	0	0	110
22 (U4)	0	0	0	0	0	0	0	0	0	0	0	-1	0	0
23 (U4)	0	0	0	0	0	0	0	0	0	0	0	1	0	92
24 (U5)	0	0	0	0	0	0	0	0	0	0	0	0	-1	0
25 (U5)	0	0	0	0	0	0	0	0	0	0	0	0	1	0.001
26 (U6)	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
27 (U6)	0	0	0	0	0	0	0	0	1	0	0	0	0	2000
28 (U7)	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
29 (U7)	0	0	0	0	0	0	0	0	0	1	0	0	0	5000
30 (U7)	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
31 (U7)	1	0	0	0	0	0	0	0	0	0	0	0	0	1
32 (U7)	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
33 (U7)	0	1	0	0	0	0	0	0	0	0	0	0	0	1
34 (U7)	0	0	-1	0	0	0	0	0	0	0	0	0	0	0
35 (U7)	0	0	1	0	0	0	0	0	0	0	0	0	0	1
36 (U7)	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
37 (U7)	0	0	0	1	0	0	0	0	0	0	0	0	0	1
38 (U7)	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
39 (U7)	0	0	0	0	1	0	0	0	0	0	0	0	0	1
40 (U7)	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
41 (U7)	0	0	0	0	0	1	0	0	0	0	0	0	0	1
42 (U7)	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
43 (U7)	0	0	0	0	0	0	1	0	0	0	0	0	0	1
44 (U7)	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
45 (U7)	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Now we solve the *dual* model (7) with objective cost vector $c = \mathbf{1}$ and using the above matrix and vector. Model (7) is in this case infeasible so, according to Theorem 3.4, the primal (1) has no complete inconsistencies. We now search for each partial inconsistency by imposing the value activating it. In practice we try to impose any possible value for each field, and every time we find a vertex for model (7) we have detected a partial inconsistency.

If we add the constraint that LEN $\geq$ 67, that corresponds to adding the following row to the above matrix,

	XPRO 1	XPRO 2	XPRO 3	XTAX 1	XTAX 2	XTAX 3	XTTAX 0-100	XINC 0-200	TTAX	INC	AGE	LEN	EPS	vector a
46	0	0	0	0	0	0	0	0	0	0	0	-1	0	-67

we obtain that (7) has a vertex solution

$$y = \{0, 0, 0, 0, 0, 0, 0.035, 0.11, 0, 0, 0, 0, 0, 0.11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10.75, 0, 12.9, 0, 53.76, 0, 0, 0, 0, 1.06\}$$

where the support is given by the 7^{th}, the 8^{th}, the 14^{th}, the 37^{th}, the 38^{th}, the 39^{th} and the 46^{th}. This means that the corresponding inequalities are forming an IIS. The partial contradiction is between the 6 inequalities corresponding to the 4 following rules, and it appears for LEN$\geq$ 67 ($46^t h$ inequality), as showed in the beginning of this Section.

M2 *income* $\geq 200 + 30\times$*length_of_career*

M3 *total_tax* $\geq 0.33\times$*income*

U2 *total_tax = tax1+tax2+tax3*

U7 *income* ≥ 0 and ≤ 5000

If we add the constraint that LEN $\geq$ 81, that corresponds to adding the following row to the above matrix,

	XPRO 1	XPRO 2	XPRO 3	XTAX 1	XTAX 2	XTAX 3	XTTAX 0-100	XINC 0-200	TTAX	INC	AGE	LEN	EPS	vector a
46	0	0	0	0	0	0	0	0	0	0	0	-1	0	-81

we obtain that (7) has a vertex solution

$$y = \{0, 0, 0, 0, 0, 0.1, 0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1\}$$

where the support is given by the 6^{th}, the 7^{th} and the 46^{th}. This means that the corresponding inequalities are forming an IIS. The partial contradiction is between the 2 inequalities corresponding to the 2 following rules, and it appears for LEN$\geq$ 81 ($46^t h$ inequality), as showed in the beginning of this Section.

M1 *income* $\leq 1000 + 20\times$*length_of_career*

M2 *income* $\geq 200 + 30\times$*length_of_career*

If we add the constraint that INC $\geq$ 2182, that corresponds to adding the following row to the above matrix,

	XPRO 1	XPRO 2	XPRO 3	XTAX 1	XTAX 2	XTAX 3	XTTAX 0-100	XINC 0-200	TTAX	INC	AGE	LEN	EPS	vector a
46	0	0	0	0	0	0	0	0	0	-1	0	0	0	-2182

we obtain that (7) has a vertex solution

$$y = \{0,0,0,0,0,0,0,16.67,0,0,0,0,0,16.67,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1667,0,2000,0,8333,0,0,0,0,5.5\}$$

where the support is given by the 8^{th}, the 14^{th}, the 37^{th}, the 39^{th}, the 41^{th} and the 46^{th}. This means that the corresponding inequalities are forming an IIS. The contradiction partial is between the 5 inequalities corresponding to the following 3 rules, and it appears for INC$\geq$ 2182 (46^th inequality), as showed in the beginning of this Section.

M3 *total_tax* $\geq 0.33\times$*income*

U2 *total_tax = tax1+tax2+tax3*

U7 *income* ≥ 0 and ≤ 5000

Therefore, the proposed fully automatic procedure was able to discover the sets of conflicting rules working only at the formal level.

References

[1] E. Amaldi, M.E. Pfetsch and L. Trotter Jr., Some structural and algorithmic properties of the maximum feasible subsystem problem, *in proc. of 10th Integer Programming and Combinatorial Optimization conference*, Lecture Notes in Computer Science 1610, Springer, 1999, 45–59.

[2] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, Massachusetts, 1997.

[3] R. Bruni, Discrete Models for Data Imputation, *Discrete Applied Mathematics* Vol. 144/1 (2004), 59-69.

[4] R. Bruni, Error Correction for Massive Data Sets, *Optimization Methods and Software*, Vol. 20/2-3 (2005), 295-314.

[5] R. Chandrasekaran, Integer programming problems for which a simple rounding type of algorithm works, In W.R. Pulleyblank, ed. *Progress in Combinatorial Optimization*, Academic Press, 1984, 101-106.

[6] J.W. Chinneck, Fast Heuristics for the Maximum Feasible Subsystem Problem, *INFORMS Journal on Computing* 13/3 (2001), 210-223.

[7] J.W. Chinneck and E.W. Dravnieks, Locating Minimal Infeasible Constraint Sets in Linear Programs. *ORSA Journal on Computing* 3 (1991), 157-168.

[8] M. Conforti, G. Cornuéjols, A. Kapoor and K. Vuskovic, Recognizing balanced 0, + or - 1 matrices, In *Proceedings 5th annual SIAM/ACM Symposium on Discrete Algorithms* (1994), 103-111.

[9] P. Fellegi and D. Holt, A Systematic Approach to Automatic edit and Imputation, *Journal of the American Statistical Association*, 71/353 (1976), 17-35.

[10] M.R. Garey and D.S. Johnson, *Computers and Intractability*, Freeman, New York, 1979.

[11] J. Gleeson and J. Ryan, Identifying Minimally Infeasible Subsystems of Inequalities, *ORSA Journal on Computing* 2/1 (1990), 61-63.

[12] O. Guieu and J.W. Chinneck, Analyzing Infeasible Mixed-Integer and Integer Linear Programs, *INFORMS Journal on Computing* 11/1 (1999), 63-77.

[13] G.L. Nemhauser and L.A. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons, Inc., New York, NY, 1999.

[14] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York, 1986.

[15] M. Tamiz, S.J. Mardle and D.F. Jones, Detecting IIS in Infeasible Linear Programs using Techniques from Goal Programming, *Computers and Operations Research* 23 (1996), 113-191.

Received: July, 2012