WP. 33
ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Ottawa, Canada, 16-18 May 2005)

Topic (iv): New and emerging methods, including automation through machine learning, imputation, evaluation of methods

DATA EDITING AND LOGIC

Supporting Paper

Submitted by the Research School of Information Sciences and Engineering,
Australian National University[1]

I Introduction

1. Automated data editing typically involves three steps:

Testing a data record against the edits.

Error localisation: Finding a smallest (weighted) set of fields that can be changed to correct the record.

Imputation: Changing those fields so as to preserve the original joint frequency distribution of the data as far as possible.

In this paper we concentrate on the second step, error localisation. Our objective is to convert the error localisation problem to a corresponding logical problem.

2. Why do this? Logic is the study of deduction, and most methods of automated error localisation can be seen as a trade-off between search and deduction, where search means systematically testing potential corrections to the erroneous record, and deduction means finding some set $D(E)$ of edits logically implied by the set E of pre-defined edits. A conversion of the error localisation problem to logic could be useful because there are many automated logic tools which could be relevant to error localisation.

3. Our overall plan is to use logic to formalise the deduction component, as opposed to the search component, of automated error localisation. Initially we have looked only at a 'pure deduction method', where no search is needed at all. An example of a pure deduction method is the Fellegi-Holt method [9].

[1] Prepared by Agnes Boskovitz, Rajeev Goré and Paul Wong (agnes.boskovitz@anu.edu.au, rajeev.gore@anu.edu.au, wongas@mail.rsise.anu.edu.au).

4. All of the pure deduction methods depend on the ‘covering set method’ which we define precisely in Section II. Here we just point out that the covering set method uses $D(E)$ to seek an error localisation solution for each record.

5. However the set of results obtained by the covering set method can include some non-solutions of the the error localisation problem and need not even include all of the error localisation solutions. That is, the covering set method is not always successful. Its success depends on the definition of the set $D(E)$. For example when $D(E)$ is constructed using the Fellegi-Holt method then the covering set method finds exactly all of the error localisation solutions. But for other definitions of $D(E)$ the covering set method fails. When the covering set method finds exactly all of the error localisation solutions we say that the method of constructing $D(E)$ is ‘(smallest weighted) covering set correctible’.

6. In this paper, we will formalise ‘covering set correctible’ in terms of logic, in Sections III and IV, each of which deals with a different class of edits. It turns out that covering set correctibility is related to error localisation just as certain corresponding logical constructs are related to each other. The corresponding logical constructs are ‘refutation completeness and soundness’, explained in Section III, and the ‘satisfiability problem’, explained in Section V. But first, in the next section, we explain some basic notions.

II Assumptions, Definitions, Notation and Examples

7. In this section we define our main notions including records, edits, deduction, the covering set method and covering set correctible. We will illustrate with some simple examples which will be used throughout the paper.

8. We assume that we are dealing with data arranged in *records*. A record R is an N-tuple $(R_1, \ldots, R_N)$, where $R_j \in A_j$, where A_j is the domain of the j-th field. We assume that N is fixed for all records under consideration. We assume that the *edits* used to specify potential errors in the data apply to one record at a time. Contrary to common practice, we will say that each edit specifies an acceptance region of the set of all possible records, whereas commonly edits are rejection regions. A record satisfies an edit when it is in the acceptance region.

9. We will classify edits into two types: *categorical*, also known as discrete; and *arithmetic*, also known as numerical or continuous. Categorical edits deal with data that has discrete structure, such as marital status, or with numerical data that can be discretised while still retaining the edits. For example the edit ‘if age < 15 then marital status $=$ never married’ is categorical because the field age can be discretised to $[0, 14]$ and $[15, \text{maximum age}]$. Arithmetic edits deal with numerical data that has not been discretised, for example $-2x_1 + x_2 \geq 0$, where x_1 and x_2 are real variables representing the values of fields 1 and 2 respectively.

10. We will work with the following two simple contrived examples.

Example II.1 (for categorical edits) *In a table of data about school children, each person is represented by a different row (or record). The table consists of the following three fields:*

age*: the age of the person*

driver*: whether the person has a driver's licence*
grade*: the person's school grade level.*

The domains of the fields are $A_{\mathsf{age}} = \{5, 6, \ldots, 20\}$*,* $A_{\mathsf{driver}} = \{\mathsf{Y}, \mathsf{N}\}$*, and* $A_{\mathsf{grade}} = \{1, 2, \ldots, 12\}$*. There are two edits specified:*

Edit C1*: A person with a driver's licence must be in at least Grade 11.*
Edit C2*: A person in Grade 7 or higher must be at least 10 years old.*

We will write $EC = \{\mathrm{C1}, \mathrm{C2}\}$*. The record* $RC = (6, \mathsf{Y}, 8)$*, with* $\mathsf{age} = 6$*,* $\mathsf{driver} = \mathsf{Y}$ *and* $\mathsf{grade} = 8$*, fails both edits.*

Example II.2 (for arithmetic edits) *A table of numerical data has three fields, and* x_1*,* x_2 *and* x_3 *are variables representing the field values. The domain of each field is the set of real numbers* $\mathbb{R}$*. There are two edits:*

$$\begin{array}{lrcrcrcl} \textbf{\textit{Edit A1}}: & -2x_1 & + & x_2 & & & \geq & 0 \\ \textbf{\textit{Edit A2}}: & x_1 & + & & & 3x_3 & \geq & 0\,. \end{array}$$

We will write $EA = \{\mathrm{A1}, \mathrm{A2}\}$*. The record* $RA = (1, 0, -1)$ *fails both edits.*

11. Given an edit set E, the edit set $D(E)$ is found via a *deduction function* D, which takes any set of edits as input and returns a set of edits as output. That is, $D : \wp(\mathsf{Edits}) \longrightarrow \wp(\mathsf{Edits})$, where $\wp(\mathsf{Edits})$ is the set of all sets of edits. Traditionally deduction functions such as D are called consequence or closure operators [14] and are defined over an entire logical language. Intuitively, a consequence operator maps an arbitrary set of logical descriptions (the assumption set) to a set of logical descriptions (the consequence set) that are implied by the assumption set.

12. Fellegi and Holt used a function that we will call D_{FH}, which gives a set of categorical edits logically implied by the categorical edit set E. Other deduction functions described in the literature include Fellegi and Holt's 'essentially new' deduction [9] (which we will call D_{ENFH}) and the deduction function defined by the Field Code Forest Algorithm of Garfinkel, Kunnathur and Liepins [11] (which we will call D_{FCF}).

Example II.3 (for categorical edits) *For the deduction function* D_{FH}*, it turns out that* $D_{\mathrm{FH}}(EC) = \{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}\}$ *where* C3 *is:*

Edit C3*: A person with a driver's licence must be at least 10 years old.*

Edit C3 *is logically implied by Edits* C1 *and* C2*. Note that the record* RC *fails* C3*.*

13. For arithmetic edits, one deduction function is defined by eliminating variables by taking positive linear combinations of inequalities, as done in the Fourier-Motzkin elimination method [10, 12]. We will call this deduction method D_{FM} (see Fellegi and Holt [9], and de Waal [7]).

Example II.4 (for arithmetic edits) *For the deduction function* D_{FM}*, it turns out that* $D_{\mathrm{FM}}(EA) = \{\mathrm{A1}, \mathrm{A2}, \mathrm{A3}\}$ *where* A3 *is obtained by adding Edit* A1 *to twice Edit* A2*:*

$\textbf{\textit{Edit A3}}: \quad x_2 + 6x_3 \geq 0\,.$ *Note that the record* RA *fails Edit* A3*.*

14. For any deduction function D, the covering set method, defined below, tries to find an error localisation solution for any record R and any edit set E. Its success depends on the properties of D.

15. We will define the covering set method and describe the relevant properties of D shortly but, first, we note that we will simplify the discussion slightly. Instead of describing methods that find all smallest weighted error localisation solutions, we will describe methods that find all error localisation solutions, or equivalently, methods that find all minimal error localisation solutions. The results are essentially the same, but there are some specialised restrictions for the 'smallest weighted' case which are just complicating side-issues.

16. The covering set method depends on an *'involved field'*: informally, a field is involved in an edit if it is mentioned in the edit.

Example II.5 *Edit* C1 *involves fields* driver *and* grade*. Edit* A1 *involves fields 1 and 2.*

17. We can now give the steps in the *covering set method*:

(a) Find the set $X(D(E), R)$, abbreviated as X, consisting of the edits of $D(E)$ failed by R.

(b) Find a set C of fields that covers X in the sense that every edit of X involves some field of C.

The objective of the covering set method is to use the covering set C to find an *error localisation* solution for the record R and the edit set E, as follows: Find a record $R' = (R'_1, \ldots, R'_N)$ such that (i) R' satisfies E, and (ii) R' differs from R at most on C, that is, if $j \notin C$ then $R'_j = R_j$. When such an R' exists, we have an error localisation solution, and we say that the field set C *yields a correction of the record* R *for edit set* E. The covering set method works for some deduction functions D but not for all, as demonstrated by the next examples.

Example II.6 (for categorical edits - unsuccessful D) *The identity deduction function* I *has* $I(E) = E$ *for each edit set* E*. The covering set method does not always work for* I*. For example,* $X(I(EC), RC) = EC$ *and one covering set is* $\{\text{grade}\}$*. However it is impossible to correct* RC *by changing only the field* grade.

Example II.7 (for categorical edits - successful D)
$X(D_{\mathrm{FH}}(EC), RC) = \{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}\}$*. The covering set method always works for the deduction function* D_{FH}*. One covering set of* X *is* $\{\text{grade}, \text{driver}\}$*, which yields a correction to* RC*, for example by changing* driver *to* N *and* grade *to* 1.

Example II.8 (for arithmetic edits - successful D) *The record* RA *fails edits* A1, A2 *and* A3*. Hence* $X(D_{\mathrm{FM}}(EA), RA) = \{\mathrm{A1}, \mathrm{A2}, \mathrm{A3}\}$*. The covering set method always works for the deduction function* D_{FM}*. One covering set of* X *is* $\{$*field 1, field 2*$\}$ *which yields a correction to* RA*, for example by changing the value of field 1 to 4 and the value of field 2 to 10.*

18. We want the covering set method to result in an error localisation solution for any covering set C of X, for any record R and any edit set E. We also want to be able to find all possible error localisation solutions by suitable choices of C. This happens for certain deduction functions, and we say that the deduction function D is *covering set correctible* if D has the following two properties:

(i) For each record R and each edit set E, each covering set C of $X(D(E), R)$ yields a correction of R for E. In this case we say that D has the *error correction guarantee* or ECG.

(ii) For each record R and each edit set E, if R' satisfies E then R' could have been obtained by the covering set method: that is, there is a C containing $\{j \in \{1,\ldots,N\} \mid R'_j \neq R_j\}$ such that C is a covering set of $X(D(E),R)$. In this case we say that D has *error correction totality* or ECT.

19. In the next sections we formalise covering set correctibility. We consider categorical edits in Section III and arithmetic edits in Section IV.

III Formalisation of covering set correctible for categorical edits

20. In order to formalise 'covering set correctible' for categorical edits we will formalise the following:

(a) edits – as logical formulae

(b) records – as truth functions

(c) satisfaction relation, i.e. 'a record satisfies an edit' – when the appropriate truth function applied to the edit is true.

(d) deduction – as already defined as a function in Section II. We will use Fellegi-Holt deduction as an example.

(e) covering set of the edits failed by a record – a property of the field set and the record, which we specify in Lemma III.1

(f) a field set yielding a correction to a record – a property of the field set and the record, which we specify in Lemma III.2.

The property 'covering set correctible' is a relationship between the above items (e) and (f). This relationship is specified in Proposition III.1, which turns out to be a strengthening of 'refutation completeness' and 'soundness', explained at the end of the section.

21. We will use classical propositional logic where formulae are built from a set of *atoms*, called Atoms, using the boolean connectives $\vee$ (or) and $\neg$ (not). Our atoms represent individual field values, as in Bruni [3]: for example, p^6_{age} stands for '$\mathsf{age} = 6$'. The set Atoms is $\{p^v_j \mid j = 1,\ldots,N, v \in A_j\}$, so that p^v_j captures that the field j has value v. A result of logic tells us that each logical formula can be represented by a set of special formulae called *clauses* which are built from atoms or negated atoms ($\neg p^v_j$) using only the propositional connective $\vee$. We represent edits as clauses. Indeed, we will see below that each edit can be represented as a *positive clause*, which is a clause built from atoms using only $\vee$, with no use of $\neg$.

Example III.1 *In order to represent the categorical edit* C1 *as a clause, we consider the two exhaustive cases* $\mathsf{driver} = \mathsf{Y}$ *and* $\mathsf{driver} = \mathsf{N}$. *When* $\mathsf{driver} = \mathsf{Y}$, *then* C1 *is satisfied exactly when* grade *is 11 or 12. When* $\mathsf{driver} = \mathsf{N}$, *then* C1 *is satisfied. Hence* C1 *can be represented as*

$$\begin{array}{lrcl} & \mathrm{C1} & = & p^{\mathsf{N}}_{\mathsf{driver}} \quad \vee \quad p^{11}_{\mathsf{grade}} \vee p^{12}_{\mathsf{grade}}\,. \\ \textit{Similarly,} & \mathrm{C2} & = & p^{10}_{\mathsf{age}} \vee \cdots \vee p^{20}_{\mathsf{age}} \quad \vee \quad p^{1}_{\mathsf{grade}} \vee \cdots \vee p^{6}_{\mathsf{grade}}\,. \end{array}$$

Note that we have doubled up the use of the notation C1 *and* C2 *- allowing* C1 *and* C2 *to refer both to edits in terms of words and edits in terms of logical clauses. The context should prevent confusion.*

22. Each record R is represented by a *truth function* $f_R : \mathsf{Atoms} \longrightarrow \{\mathsf{true}, \mathsf{false}\}$ which can be extended uniquely to a truth function on any clause, including on edits. The record R *satisfies* the edit e when $f_R(e) = \mathsf{true}$. There are restrictions on which truth functions can serve to represent records because records have specific constraints: each component of each record takes exactly one value. This means that a truth function must map to the truth value true each of the following clauses, called *'axioms'*:

Axiom 1 : $\neg p_j^v \vee \neg p_j^w$, for all $j = 1, \ldots, N$ and for $v \neq w$. (Each field of a record has at most one value.)

Axiom 2 : $\bigvee_{v \in A_j} p_j^v$, for all $j = 1, \ldots, N$. (Each field of a record has at least one value.)

Example III.2 *For* $RC = (6, \mathsf{Y}, 8)$*, the corresponding truth function* f_{RC} *is defined by* $f_{RC}(p^6_{\mathsf{age}}) = f_{RC}(p^{\mathsf{Y}}_{\mathsf{driver}}) = f_{RC}(p^8_{\mathsf{grade}}) = \mathsf{true}$*. For all other atoms* p*,* $f_{RC}(p) = \mathsf{false}$*. For the edits,* $f_{RC}(\mathrm{C1}) = f_{RC}(\mathrm{C2}) = \mathsf{false}$*, meaning that the record fails both edits.*

23. By Axiom 2, any negated atom $\neg p_j^w$ always has the same truth value as the positive clause $\bigvee\{p_j^v \mid v \in A_j \setminus \{w\}\}$. By replacing each negated atom in clause e_1 by its corresponding positive clause, we obtain a positive clause e_2 with the same truth value as e_1. Hence we can represent each edit as a positive clause. In general an edit has the form

$$e = \bigvee\{p_j^v \mid j = 1, \ldots, N \text{ and } v \in V_j^e\},$$

where $V_j^e \subseteq A_j$. If $V_j^e = \emptyset$ for all j then e is the *empty clause*, written as $\Box$, and called 'box', which is assigned to false by every truth function, and which is the undesirable edit that fails all records. If $V_j^e \neq \emptyset$ then we say that the edit e involves field j.

24. The three deduction functions described in the literature (D_{FH}, D_{ENFH} and D_{FCF}) can be expressed in terms of our logical formalisation. For example the function D_{FH} is defined as follows. Given a set E of edits, let E' be any subset of E and let i be any field. Define the D_{FH}-deduced edit on E' with generating field i as $\mathrm{FHD}(i, E')$ where

$$\mathrm{FHD}(i, E') = \bigvee\Big\{p_j^v \mid j \in \{1, \ldots, N\} \setminus \{i\}, v \in \bigcup_{e \in E'} V_j^e\Big\} \quad \vee \quad \bigvee\Big\{p_i^v \mid v \in \bigcap_{e \in E'} V_i^e\Big\}.$$

This is identical to Fellegi and Holt's definition, except it is in terms of the acceptance region rather than the rejection region. Fellegi and Holt's 'normal edits' are the same as our positive clauses. Given a starting set E of edits, the edit $\mathrm{FHD}(i, E')$ can be found for all subsets E' of E for all generating fields i. The newly found edits can then be added to E and the process repeated until no new edits are generated. At any time, we can remove any edit that is a superset, as a set of atoms, of some other edit in E. Such removed edits are called 'dominated' in the editing literature (Garfinkel, Kunnathur and Liepins [11], Winkler [15]), and 'subsumed' in the logic literature [4]. The process will eventually terminate if the field domains are finite. The end result is the edit set $D_{\mathrm{FH}}(E)$.

Example III.3 $\mathrm{FHD}(\mathsf{grade}, EC) = p^{10}_{\mathsf{age}} \vee \cdots \vee p^{20}_{\mathsf{age}} \vee p^{\mathsf{N}}_{\mathsf{driver}}$ *. This clause is a logical representation of Edit* C3*, and is the only positive clause that can be generated. Hence,* $D_{\mathrm{FH}}(EC) = \{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}\}$ *.*

25. It turns out that D_{FH}, D_{ENFH} and D_{FCF} are all subfunctions of a deduction function, important to logic, called *'resolution'*. The details are given in Boskovitz et al [2].

26. Hence certain subfunctions of resolution can be used to do error localisation by the covering set method. The question is: which other subfunctions of resolution and which other deduction functions can also be used successfully to do error localisation by the covering set method? To answer this we need to formalise the covering set method in logic, through the concepts of:

(i) a covering set $X(D(E), R)$, and

(ii) a field set C yielding a correction of record R for edit set E.

The property 'covering set correctible' can then be expressed in terms of these two.

27. We have specified the two concepts above using a set constructed by the Davis-Putnam-Logemann-Loveland (DPLL) *splitting rule* [6, 5], important in logic. This rule can be used when we are given a partial truth function $f_{R,Z}$ (for a record R and a field set Z) defined by $f_R(p_j^v)$ if $j \in Z$ and undefined otherwise. The partial truth function $f_{R,Z}$ can be extended to a partial truth function on all clauses. Given a partial truth function $f_{R,Z}$, the DPLL splitting rule reduces an edit set S to an edit set[2] $S[R, Z]$. The reduction proceeds by removing some edits from S and removing atoms from other edits in S. Specifically, if $s \in S$ and $f_{R,Z}(s) = \mathsf{true}$ then s is excluded from $S[R, Z]$; s is excluded exactly when s contains an atom p_j^v with $f_{R,Z}(p_j^v) = \mathsf{true}$. If $s \in S$ and $f_{R,Z}(s)$ is false or undefined, then any atom p_j^v in s with $f_{R,Z}(p_j^v) = \mathsf{false}$ is deleted from s and what is left of s is put in $S[R, Z]$. Such deleted atoms must have $j \in Z$. More formally, $S[R, Z]$ is defined by:

$$S[R, Z] = \left\{ s \setminus \{p_j^v \mid j \in Z, v \in A_j\} \;\middle|\; s \in S \text{ and } f_{R,Z}(s) \neq \mathsf{true} \right\}.$$

Example III.4 *Let $S = EC$, $R = RC$, and $Z = ZC = \{\mathsf{age}, \mathsf{driver}\}$. Then*

$$EC[RC, ZC] \;=\; \{p^{11}_{\mathsf{grade}} \vee p^{12}_{\mathsf{grade}} \;,\; p^{1}_{\mathsf{grade}} \vee \cdots \vee p^{6}_{\mathsf{grade}}\}\,.$$

Note that $EC[RC, ZC]$ is unsatisfiable, that is, there is no truth function that takes the value true *for both clauses in $EC[RC, ZC]$.*

28. The set $S[R, Z]$ can contain the empty clause $\Box$. This happens if there is an s in S with $f_{R,Z}(s) = \mathsf{false}$ - that is, every atom p_j^v in s has $f_{R,Z}(p_j^v) = \mathsf{false}$. In this case all the atoms are deleted from s, leaving the empty clause.

Example III.5 *Let $S = D_{\mathrm{FH}}(EC) = EC \cup \{\mathrm{C3}\}$, $R = RC$, and $Z = ZC$. Then*

$$(D_{\mathrm{FH}}(EC))[RC, ZC] \;=\; \{p^{11}_{\mathsf{grade}} \vee p^{12}_{\mathsf{grade}} \;,\; p^{1}_{\mathsf{grade}} \vee \cdots \vee p^{6}_{\mathsf{grade}} \;,\; \Box\}\,.$$

29. We can now use certain reduced sets of clauses to give a logical formalisation of the concepts 'covering set' and 'yields a correction', in the following lemmas.

Lemma III.1 (Formalisation of 'covering set') *The field set C is a covering set of the edit set $X(S, R)$ (= the edits of S failed by R) if and only if $\Box \notin S[R, \overline{C}]$, where $\overline{C}$ is the complement of C.*

Proof in summary: C is not a covering set $\Leftrightarrow$ there is an edit in S failed by R and which does not involve C $\Leftrightarrow$ $\Box \in S[R, \overline{C}]$.

[2] Note that in another paper [1], we used a different notation. Instead of $S[R, Z]$, we used $S[R_i \mid i \notin \overline{Z}]$.

Example III.6 *The set $CC = \{\mathsf{grade}\}$ is a covering set of $X(EC, RC)$. Since $\overline{CC} = ZC$, from Example III.4, $\Box \notin EC[RC, \overline{CC}]$.*

Lemma III.2 (Formalisation of 'yields a correction') *The field set C yields a correction of the record R for the edit set S if and only if there is some truth function which assigns all clauses in $S[R, \overline{C}]$ to true (or more briefly 'if and only if $S[R, \overline{C}]$ is satisfiable').*

Proof in summary: C yields a correction R' of R for $S \;\Leftrightarrow\; f_{R',C}$ satisfies $S[R, \overline{C}]$.

Example III.7 *As seen in Example II.6, the set CC does not yield a correction of RC for EC. From Example III.4, $EC[RC, \overline{CC}]$ is not satisfiable.*

30. Using Lemmas (III.1) and (III.2) we can formalise 'covering set correctible':

Proposition III.1 (Formalisation of 'covering set correctible') *The deduction function D is covering set correctible if and only if the following statement holds: For each edit set E, field set Z, and record R*

$$\Box \notin (D(E))[R, Z] \;\Leftrightarrow\; E[R, Z] \textit{ is satisfiable.}$$

Proof in summary: use the lemmas, and $C = \overline{Z}$ = covering set of $X(D(E), R)$.

Example III.8 *Let D be the identity function I, which is not covering set correctible. For example, let $Z = ZC = \overline{CC}$. Then from Example III.4, $\Box \notin (I(E))[RC, ZC]$, but $EC[RC, ZC]$ is unsatisfiable.*

31. Using the lemmas, the error correction guarantee (ECG), defined in paragraph 18(i), can be seen to be the forward direction of the condition in Proposition III.1, that is:

$$\Box \notin (D(E))[R, Z] \Rightarrow E[R, Z] \text{ is satisfiable.}$$

The ECG is a strengthening of *refutation completeness*, which is the following property of D. For each edit set E,

$$\Box \notin D(E) \Rightarrow \; E \text{ is satisfiable.}$$

To say that D has refutation completeness is to say that the absence of $\Box$ in $D(E)$ guarantees that some record satisfies E. This is in parallel with the ECG: to say that D has the ECG is to say that the absence of $\Box$ in $(D(E))[R, Z]$ guarantees that a particular type of record satisfies E, namely some record obtained by changing R on at most $\overline{Z}$.

32. Using the lemmas, error correction totality (ECT), defined in paragraph 18(ii), can be seen to be the backward direction of the condition in Proposition III.1, that is:

$$E[R, Z] \text{ is satisfiable} \Rightarrow \Box \notin (D(E))[R, Z].$$

ECT is a strengthening of *soundness*, which is the following property of D. For each edit set E,

$$E \text{ is satisfiable } \;\Rightarrow\; \Box \notin D(E).$$

To say that D is sound is to say that $D(E)$ can be used to find every E that has a satisfying record. This is in parallel with ECT: to say that D has ECT is to say that $(D(E))[R, Z]$ can be used to find every E that has a satisfying record of a particular type, namely a record obtained by changing R on at most $\overline{Z}$.

33. What does the above mean? It characterises the underlying properties that a deduction function must have if it is to be covering set correctible. The underlying property is systematically stronger than refutation completeness and soundness. These two latter properties underpin many automated logic tools. The corresponding properties, ECG and ECT, expressed in terms of the logical formalisation, would underpin any modification for error localisation of the logical tools.

34. We now explore ECG and ECT for arithmetic edits.

IV Formalisation of covering set correctible for arithmetic edits

35. Our formalisation of covering sets for arithmetic edits follows the same steps as for categorical edits. We formalise edits, records, satisfaction relation, deduction, covering sets and correction. We find, as for categorical edits, that 'covering set correctible' is a strengthening of refutation completeness and soundness.

36. The general form of an arithmetic edit is

$$\sum_{j=1}^{N} a_j x_j \quad \geq \quad b, \text{ where } a_j, b \in \mathbb{R}\,.$$

37. Each record R is represented as an N-tuple $(R_1, \ldots, R_N)$ of reals. The record R satisfies the edit $\sum_{j=1}^{N} a_j x_j \geq b$ when $\sum_{j=1}^{N} a_j R_j \geq b$.

38. If the edit $\sum_{j=1}^{N} a_j x_j \geq b$ has $a_j = 0$ for all j and has $b \in \mathbb{R}^+$ (the positive reals), then the edit simplifies to $0 \geq b$ with $b \in \mathbb{R}^+$. Since no record satisfies this edit, we write it as $\Box$, the equivalence class of edits failed by all records.

39. An example of a deduction function is D_{FM}. We define D_{FM} as follows. First write the edit set E as $A\mathbf{x} \geq \mathbf{b}$, where $\mathbf{x}$ is an N-dimensional column vector and each component of $A\mathbf{x} \geq \mathbf{b}$ represents one edit. Then define $D_{\mathrm{FM}}(E)$ to be the set of those non-negative linear combinations of the edits of E where at least one variable is eliminated. That is, $D_{\mathrm{FM}}(E)$ is a set of edits of the form $\mathbf{y}^T A\mathbf{x} \geq \mathbf{y}^T\mathbf{b}$. De Waal [7, page 47] explains why D_{FM} is covering set correctible.

Example IV.1 $D_{\mathrm{FM}}(EA) = \{\mathrm{A}1, \mathrm{A}2, \mathrm{A}3\}$.

40. In order to formalise 'covering set correctible', we construct a set via the analogue, for inequalities, of the DPLL splitting rule. Given an edit set S, a record R and a field set Z, we define $S[R, Z]$ to be the set of edits of the form

$$\textstyle\sum_{j \notin Z} a_j x_j \geq b - \sum_{j \in Z} a_j R_j$$

where the inequality $\sum_{j=1}^{N} a_j x_j \geq b$ is an edit in S.

Example IV.2 *Let $ZA = \{$field 2, field 3$\}$. Then $EA[RA, ZA] = \{-2x_1 \geq 0,\ x_1 \geq 3\}$ and $(D_{\mathrm{FM}}(EA))[RA, ZA] = \{-2x_1 \geq 0,\ x_1 \geq 3,\ \Box\}$. Note that both edit sets are unsatisfiable.*

41. With these definitions, the lemmas and the proposition of Section III.4 hold also for arithmetic edits. This means that the property 'covering set correctible' for arithmetic edits is a strengthening of refutation completeness and soundness, just as it is for categorical edits. It is interesting that the underlying reason for the success of the Fourier-Motzkin method for solving inequalities can be seen as Farkas' Lemma, which is equivalent to refutation completeness and soundness.

Example IV.3 *The deduction function D_{FM} is covering set correctible. By way of an example supporting the lemmas and proposition, consider the field set $\overline{ZA} = \{\text{field } 1\}$. The set $\overline{ZA}$ does not cover $X(D_{\mathrm{FM}}(EA), RA)$, nor does it yield a correction of RA for EA. By the lemmas, this last sentence is equivalent to (i) $\Box \in (D_{\mathrm{FM}}(EA))[RA, ZA]$, and (ii) $EA[RA, ZA]$ is unsatisfiable, supporting the condition of the proposition.*

V Parallel with SAT

42. *Propositional satisfiability* (known as SAT) is the problem of deciding whether a given set of clauses is satisfiable. The error localisation problem is an extension of SAT: rather than just deciding whether a set of edits and axioms is satisfiable, we seek to decide whether each field set C yields a correction to the record.

43. Just as the error localisation problem and the SAT problem are related, so are their solutions. Just as the error localisation problem can be solved by a full deduction method, so can the SAT problem. An example of a full deduction solution for SAT is directional resolution, described in Dechter [8].

44. Not only are the solutions to the two problems related, but so are the reasons that the solutions succeed. The covering set method succeeds exactly when the deduction function has the properties ECG and ECT. These properties are strengthenings of refutation completeness and soundness - exactly the properties causing the directional resolution method for SAT to succeed. Table 1 summarises these relationships.

Table 1: Parallels between error localisation and satisfiability

Problem	*Error Localisation*	*Satisfiability*
Solution Method	covering set method	directional resolution
Property needed for method to find solution	covering set correctibility	refutation completeness and soundness

45. Both SAT and error localisation have other more common methods of solution than the full deduction methods. Both are more commonly solved using a combination of search and deduction. The search methods of SAT include the procedure of Davis, Putnam, Logemann and Loveland [5]. The search methods of error localisation include cutting plane techniques, and branch and bound techniques (Garfinkel, Kunnathur, Liepins [11], Ragsdale and McKeown [13], de Waal [7]). It seems likely that there will be parallels between the search techniques for error localisation and the search techniques for SAT, which could shed light on both problems.

VI Conclusion

46. This paper has presented the beginnings of a theoretical logical framework for analysing the error localisation problem. The main aspects are listed below.

(a) The generation of new edits can be seen as logical deduction, where information implicit in a set of edits is extracted as conclusions.

(b) The covering set method for error localisation is successful exactly when the deduction function has covering set correctibility, which is a strengthening of refutation completeness and soundness.

(c) The error localisation problem is a strengthening of the satisfiability problem.

(d) The directional resolution method of solving the satisfiability problem depends on refutation completeness and soundness in just the same way as the covering set method of solving the error localisation problem depends on covering set correctibility. Table 1 displays these parallels.

(e) The same results hold for categorical and arithmetic edits.

47. Logic gives two benefits. Firstly, it gives an alternative way of analysing the problem and thus potentially gives new insights as in points (a) to (e) above. Secondly, its collection of sophisticated automated tools could potentially be modified to use covering set correctibility for solving error localisation problems.

References

[1] Agnes Boskovitz and Rajeev Goré. Automatic data editing: a framework from logic. In *55th Session of the International Statistical Institute*, 5–12 April 2005.

[2] Agnes Boskovitz, Rajeev Goré, and Markus Hegland. A logical formalisation of the Fellegi-Holt method of data cleaning. In Michael R. Berthold et al, editors, *Advances in Intelligent Data Analysis, IDA 2003*, volume 2810 of *Lecture Notes in Computer Science*, pages 554–565. Springer-Verlag, August 2003.

[3] Renato Bruni and Antonio Sassano. Errors detection and correction in large scale data collecting. In Frank Hoffmann et al, editors, *Advances in Intelligent Data Analysis, IDA 2001*, volume 2189 of *Lecture Notes in Computer Science*, pages 84–94. Springer-Verlag, September 2001.

[4] Chin-Liang Chang and Richard Char-Tung Lee. *Symbolic logic and mechanical theorem proving.* Academic Press, 1973.

[5] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, July 1962.

[6] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the Association for Computing Machinery*, 7(3):201–215, July 1960.

[7] Ton de Waal. *Processing of Erroneous and Unsafe Data.* PhD thesis, Erasmus University, Erasmus Research Institute of Management (ERIM), Rotterdam, 2003.

[8] Rina Dechter and Irina Rish. Directional resolution: The Davis-Putnam Procedure, revisited. In J. Doyle et al, editors, *Principles of Knowledge Representation and Reasoning, KR'94*, pages 134–145. Kaufmann, 1994.

[9] I. P. Fellegi and D. Holt. A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association*, 71(353):17–35, March 1976.

[10] Jean Baptiste Joseph Fourier. Solution d'une question particulière du calcul des inégalités. In *Oeuvres II.* Publiés en 1888-90 par les soins de G. Darboux sous les auspices du Ministère de l'instruction publique, Paris, 1826.

[11] R. S. Garfinkel, A. S. Kunnathur, and G. E. Liepins. Optimal imputation of erroneous data: Categorical data, general edits. *Operations Research*, 34(5):744–751, Sep.–Oct. 1986.

[12] T. S. Motzkin, H. Raffa, G. L. Thompson, and R. M. Thrall. The double description method. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume II, pages 51–73. Princeton University Press, 1953.

[13] Cliff T. Ragsdale and Patrick G. McKeown. On solving the continuous data editing problem. *Computers & Operations Research*, 23(3), 263–273 1996.

[14] A. Tarski. *Logic, Semantics, Metamathematics: Papers from 1923 to 1938.* Oxford University Press, 1956.

[15] William E. Winkler. Editing discrete data. Statistical Research Report Series, RR97/04, U.S. Bureau of the Census, 1997.