182
9-24-80 JAC

Pr. 1762

ornl

ORNL/TM-7156

MASTER

**OAK
RIDGE
NATIONAL
LABORATORY**

**UNION
CARBIDE**

**Refinements to the Boolean
Approach to Automatic
Data Editing**

G. E. Liepins

**OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A03; Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-7156
Dist. Category UC-32

Contract No. W-7405-eng-26

Energy Division

REFINEMENTS TO THE BOOLEAN APPROACH TO
AUTOMATIC DATA EDITING

G. E. Liepins

Date Published - September 1980

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

Copies of the following related reports completed at Oak Ridge National Laboratory are available from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

<u>Title</u>	<u>Author(s)</u>	<u>Document Number</u>	<u>Date</u>
"A Rigorous Systematic Approach to Automatic Data Editing and Its Statistical Basis"	G. E. Liepins	ORNL/TM-7126	1980
"Sampling Theory Methodology Applicable to Data Validation Studies"	M. R. Chernick	ORNL/TM-7084	1980
"The Influence Function and Its Application to Data Validation"	M. R. Chernick	ORNL/TM-6871	1979

CONTENTS

	Page
ACKNOWLEDGMENTS	v
ABSTRACT	1
1. INTRODUCTION	1
2. THEORETICAL DETAILS OF THE FELLEGI-HOLT APPROACH	3
3. THE SET-COVERING PROBLEM	6
4. IMPLIED EDIT GENERATION ALGORITHM	9
5. NORMAL EDIT GENERATING ALGORITHM	19
6. CONCLUSIONS	22
APPENDIX	27
REFERENCES	33

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

ACKNOWLEDGMENTS

The author is indebted to Robert S. Garfinkel of the Management Science Department of the University of Tennessee for numerous lively discussions of automatic data editing and for his careful reading of earlier drafts. Any remaining oversights are mine alone.

This work is sponsored by DOE's Office of Energy Information Validation.

REFINEMENTS TO THE BOOLEAN APPROACH TO
AUTOMATIC DATA EDITING

G. E. Liepins

ABSTRACT

Automatic data editing consists of three components: (1) identification of erroneous records, (2) identification of the most likely erroneous fields within an erroneous record (fields to impute), and (3) assignment of acceptable values to failing records. Moreover, the types of data considered naturally fall into three categories: (1) coded (categorical) data, (2) continuous data, and (3) mixed data (both coded and continuous). For the case of coded data, a natural way to approach automatic data editing is commonly referred to as the Boolean approach, first developed by Fellegi and Holt. For the fields to impute problem, central to the operation of the Fellegi-Holt approach is the explicit recognition of certain implied edits; Fellegi and Holt originally required a "complete" set of edits, and their algorithm to generate this complete set has occasionally had the distinct disadvantage of failing to converge within reasonable time. The primary result of this paper is an algorithm which significantly prunes the Fellegi-Holt edit generation process, yet nonetheless, generates a "sufficient collection" of implied edits, adequate for the solution of the fields to impute problem.

1. INTRODUCTION

Automatic data editing is the computerized identification and correction (optional) of data errors. As long as only the detection of data errors by well established constraint conditions (edits) is involved, the process is straightforward. However, when the original respondent submitting the erroneous data record is unavailable to correct the errors, or some other reason precludes deterministic correction by tracing the records' genesis, it becomes necessary to either modify the erroneous records strictly on the basis of information available in the data base

or to discard the records. In a very real sense, deletion of records reflects a tacit assumption about the data. The hope is that estimates will not be biased by such deletions. In certain cases this may indeed be true, but if there is any correlation between the cause(s) of the errors and the estimated quantities, then deletion may no longer be the best policy.

This paper deals with the first step of modifying erroneous records, the identification of the most probable wrong components (fields) of an erroneous multivariate record, or in more technical terms, the fields to impute problem. This problem is relevant because although the edits indicate that an error is present, they generally do not specify the cause of the error. As a trivial example, the edit $x + y \leq 4$ indicates that the record (2,3) is in error, but does not specify which field(s) is (are) incorrect. The scope of this paper is restricted even further, insofar as its objective is to demonstrate how to determine the minimum number of (weighted) changes necessary to make acceptable an erroneous coded data record. Formally, the problem is a constrained discrete optimization problem with a fixed objective function: let x be a data record with components (fields) x_i , y be a correction factor to be determined, and $c_i > 0$ externally given constants. Then the fields to impute problem can be expressed as

$$\text{minimize } \sum c_i \delta(y_i)$$

subject to $x + y$ is an acceptable record,

$$\delta(y_i) = \begin{cases} 0 & y_i = 0 \\ 1 & y_i \neq 0 \end{cases} .$$

Only the Boolean approach for the discrete variable problem is considered, and the principal result is an algorithm which, in certain cases, significantly prunes the Fellegi-Holt¹ implied edit generation algorithm.

2. THEORETICAL DETAILS OF THE FELLEGI-HOLT APPROACH

The following section presents in a mathematically rigorous fashion the central ideas of the Fellegi-Holt approach to automatic data editing, including the usefulness of the sufficient class of implied edits to the fields to impute problem. This theory will be presented exclusively for coded data; the detailed proofs are given in certain cases in Fellegi-Holt¹ and in other cases in Liepins.³

Def. 1. The (univariately) feasible values for field i are A_i .

Def. 2. Let each record of the data set have n fields. Then the data space is defined to be the Cartesian product

$$\prod_{i=1}^n A_i .$$

Def. 3. An edit is a consistency check which defines a failure region. An edit e is in normal form if and only if it can be expressed as a Cartesian product, that is,

$$e = \prod_{i=1}^n \pi_i(e) ,$$

where $\pi_i()$ is the projection on the i^{th} field. Often an edit e will be written as

$$e = \prod_{i=1}^n A_i^r$$

where clearly $A_i^r = \pi_i(e)$, and the superscript serves only to distinguish that A_i^r can be a proper subset of A_i .

Def. 4. A normal form edit e is *essential* if and only if $A_i^r = A_i$ for at least one of the fields. Such a field is called an essential field.

Def. 5. A field i *enters* the normal form edit e if and only if it is not an essential field of e .

Def. 6. Let $\{e_i\}$ be a collection of edits. Then an edit e is an *implied edit* (by collection $\{e_i\}$) if and only if $e \subset \cup e_i$, where the union and inclusion are taken in the point-set sense.

Example 1

Consider a data space $A_1 \times \dots \times A_6$ where

$$\begin{aligned} A_1 &= \{0,1\} & A_4 &= \{0,1,2,3\} \\ A_2 &= \{0,1,2\} & A_5 &= \{0,1,2\} \\ A_3 &= \{0,1\} & A_6 &= \{0,1,2,3\} \end{aligned} ,$$

and explicit edits

$$\begin{aligned} e_1 &= A_1 \times \{0,1\} \times \{0\} \times A_4 \times \{0,1\} \times A_6 \\ e_2 &= \{1\} \times A_2 \times \{1\} \times \{0,1\} \times A_5 \times \{2,3\} \\ e_3 &= \{0\} \times \{1,2\} \times A_3 \times \{1,2,3\} \times A_5 \times A_6 \\ e_4 &= A_1 \times \{0,2\} \times A_3 \times A_4 \times A_5 \times \{0,1\} \\ e_5 &= \{1\} \times A_2 \times A_3 \times \{0\} \times \{1,2\} \times A_6 \end{aligned} .$$

In this example, fields 1, 4, and 6 are essential fields of edit e_1 . Moreover, it can easily be verified, for example, that $e_6 = A_1 \times \{1,2\} \times \{1\} \times \{1\} \times A_5 \times \{2,3\}$ is an implied edit.

The natural question about the expressibility of any edit in normal form is answered by lemma 1.

Lemma 1. Let each A_i be bounded. Then every edit can be expressed as the finite union of normal form edits.

Henceforth, only normal form edits will be considered. As alluded to earlier, certain implied edits are critical to the functioning of the Fellegi-Holt approach to data editing. Lemma 2 provides a means for constructing these, and is essentially the lemma on p. 23 of the Fellegi-Holt¹ paper.

Lemma 2. Let S be an index set, and let

$$e_r = \prod_{i=1}^n A_i^r, \quad r \in S$$

be edits. Assume that

$$\emptyset \neq A_i^* = \bigcap_{r \in S} A_i^r \text{ for } i \neq j \text{ and } r \in S, \quad ,$$

and that

$$0 \neq A_j^* = \bigcup_{r \in S} A_j^r$$

Then

$$e^* = \prod_{i=1}^n A_i^*$$

is an implied edit.

Def. 7. The field indexed by j in lemma 2 is the *generating field* for e^* .

Example 2

The edit e_6 (of example 1) can be generated from edits e_2 and e_3 with field 1, the generating field.

Ultimately, it will be necessary to know that the method of lemma 2 is adequate to generate a sufficient collection of implied edits to solve the fields to impute problem. Lemma 3 (a minor modification of theorem 2 of Fellegi-Holt) provides the first step of the logical argument which culminates in theorem 4 and justifies the above assertion.

Lemma 3. Let e be any implied edit. Then selected application of the method of lemma 2 on the explicit edits and selected repeated application of the method on the resultant edits will yield an edit e^* which satisfies $e \in e^*$.

Def. 8. A set S of edits is *sufficient* if and only if for each data element x and each edit (either explicit or implied) with exactly fields $i_1, \dots, i_k$ essential and $x \in e$, there exists an edit $\hat{e} \in S$ satisfying $x \in \hat{e}$ and at least fields $i_1, \dots, i_k$ are essential.

The usefulness of a sufficient set of edits is stated in theorem 1, which is a modification of theorem 1 of Fellegi and Holt. The original theorem was stated in terms of all edits, that is, all explicit and implied edits. The present theorem is slightly stronger insofar as it depends only on a sufficient collection of edits.

Theorem 1. Assume that there is at least one point

$$x \in \prod_{i=1}^n A_i$$

not failing any edit. Furthermore let E be the subset of a sufficient set of edits failed by the point x^0 . Let I be a (weighted) minimal set of field indices satisfying the property that if $e \in E$, then there exists $i \in I$ such that the field i enters the edit e , that is, $\forall e \in E, \exists i, i \in I$ which enters e , and $\sum_{i \in I} c_i$ is minimized. Then this set I is precisely a solution (possibly non-unique) to the fields to impute problem insofar as there exists a point $\hat{x}$ which passes all edits and satisfies $\hat{x}_i = x_i^0$ for $i \notin I$, $\hat{x}_i \neq x_i^0$ for $i \in I$, and furthermore that no smaller (weighted) number of changes can be made to make the record x^0 acceptable.

3. THE SET-COVERING PROBLEM

Example 3

Consider again the data space of example 1, and the five edits:

$$\begin{aligned} e_1 &= A_1 \times \{0,1\} \times \{0\} \times A_4 \times \{0,1\} \times A_6 \\ e_2 &= \{1\} \times A_2 \times \{1\} \times \{0,1\} \times A_5 \times \{2,3\} \\ e_3 &= \{0\} \times \{1,2\} \times A_3 \times \{1,2,3\} \times A_5 \times A_6 \\ e_4 &= A_1 \times \{0,2\} \times A_3 \times A_4 \times A_5 \times \{0,1\} \\ e_5 &= \{1\} \times A_2 \times A_3 \times \{0\} \times \{1,2\} \times A_6 \end{aligned}$$

Let the current record be $(1,0,0,0,1,0)$. This record fails edits e_1 , e_4 , and e_5 . Let the weights c_i (as in theorem 1) be equal to unity. Accordingly, if these five explicit edits were sufficient to determine the fields to impute, one would solve the following set-covering problem to identify these fields:

$$\begin{array}{l} \min \sum w_i \\ \text{subject to } \begin{vmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{vmatrix} w \geq \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix} w_i = 0,1 \end{array}$$

This problem corresponds to finding the minimal collection of fields such that for each edit failed, at least one field enters the edit. In particular, the first row of the matrix indicates that the first field does not enter the edit e_1 , the second and third do, and with the exception of field 5 the remaining ones do not. One minimal solution is $w = (1,1,0,0,0,0)$, which would suggest in the case of a sufficient set of edits that there would be a change of exactly the entries in the first two fields which ensures that the record passes all edits. In fact, since $\{e_1, \dots, e_5\}$ is not a sufficient set of edits, it develops that no such change leads to the satisfaction of these edits. Loosely speaking, this indicates that there is an implied edit which contains the original record and for which the first two fields are essential; this observation provides the basis for an algorithm which is an alternative to that of Fellegi-Holt in that it does not require the preliminary determination of a sufficient collection of implied edits (Garfinkel²). On the other hand, one could determine the following sufficient collection of edits:

$$\begin{aligned} &\{e_1, \dots, e_5\} , \\ e_6 &= A_1 \times \{1,2\} \times \{1\} \times \{1\} \times A_5 \times \{2,3\} , \\ e_7 &= A_1 \times A_2 \times \{0\} \times A_4 \times \{0,1\} \times \{0,1\} , \\ e_8 &= A_1 \times \{1\} \times A_3 \times \{1\} \times \{0,1\} \times \{2,3\} , \\ e_9 &= A_1 \times \{2\} \times \{1\} \times \{1\} \times A_5 \times A_6 , \\ e_{10} &= \{0\} \times A_2 \times \{0\} \times \{1,2,3\} \times \{0,1\} \times A_6 , \\ e_{11} &= \{0\} \times A_2 \times A_3 \times \{1,2,3\} \times A_5 \times \{0,1\} , \\ e_{12} &= \{1\} \times A_2 \times \{0\} \times \{0\} \times A_5 \times \{0,1\} , \\ e_{13} &= \{1\} \times \{0,1\} \times A_3 \times \{0,1\} \times \{0,1\} \times \{2,3\} , \\ e_{14} &= \{1\} \times \{0,1\} \times A_3 \times \{0\} \times A_5 \times \{2,3\} , \\ e_{15} &= \{1\} \times \{0,1\} \times \{0\} \times \{0\} \times A_5 \times A_6 , \\ e_{16} &= \{1\} \times \{0,2\} \times \{1\} \times \{0,1\} \times A_5 \times A_6 , \\ e_{17} &= \{1\} \times \{0\} \times A_3 \times \{0\} \times A_5 \times A_6 , \\ e_{18} &= \{1\} \times \{0\} \times A_3 \times \{0,1\} \times \{0,1\} \times A_6 . \end{aligned}$$

(The efficient determination of such a sufficient set is the principal result of this paper.)

In terms of this collection, the record (1,0,0,0,1,0) fails edits $e_1, e_4, e_5, e_7, e_{12}, e_{15}, e_{17}$, and e_{18} . Thus, the fields to impute problem becomes

$$\begin{array}{rcl} & \min \sum w_i & \\ \text{subject to} & \begin{array}{|c|} \hline 0 \ 1 \ 1 \ 0 \ 1 \ 0 \\ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \\ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \\ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \\ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \\ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \\ \hline \end{array} & w_i \geq \begin{array}{|c|} \hline 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \hline \end{array}, w_i = 0,1 \end{array}$$

One solution is $w_1 = w_2 = w_3 = 1$, and $w_4 = w_5 = w_6 = 0$. For example, the record (0,1,1,0,1,0) passes all edits.

In general, with use of the Fellegi-Holt results, the fields to impute problem:

$$\begin{array}{ll} \text{given } c_i > 0 & \\ \text{minimize } \sum c_i \delta(y_i) & \\ \text{subject to } x + y \text{ is an acceptable record} & (1) \\ \delta(y_i) = \begin{cases} 0 & y_i = 0 \\ 1 & y_i \neq 0 \end{cases} & \end{array}$$

can be written as a set-covering problem (2-3) (see Garfinkel and Nemhauser³),

$$\min \sum c_i w_i \quad (2)$$

$$\text{subject to } Mw \geq 1, w_i = 0,1 \quad (3)$$

where M is the failed edit matrix (local to the erroneous record x) with rows indexed by the failed edits, columns by the fields, 1's corresponding to fields that enter, and 0's to essential fields.

4. IMPLIED EDIT GENERATION ALGORITHM

The primary drawback to straightforward use of the Fellegi-Holt algorithm (lemma 2) to generate a sufficient set of edits is that if the cardinality of the set of explicit edits is m , the algorithm may require an exponential function of m number of applications to generate all the necessary implied edits. Of course, what would be desirable would be to consider only combinations which yield new essential edits. On page 29 of their paper, Fellegi and Holt¹ give three rules to improve their algorithm. These are contained in algorithm 1 described below. To facilitate understanding of the algorithm, certain concepts and definitions are given by example.

Example 4

Consider a data set with four fields. The field index ordered tree is given below:

	Branch 1			Branch 2		Branch 3	Branch 4
Root node (Level 1)		1		2		3	4
Level 2	12	13	14	23	24	34	
Level 3	123	124	134	234			
Level 4	1234						

1. The order of traversal processes branch i before $i + 1$, the element i^* before the element i^{**} , and the element ij^{**} before the element $i(j+1)^*$. Thus, the order of traversal is 1; 1,2; 1,2,3; 1,2,3,4; 1,2,4; 1,3; 1,3,4; 1,4;

2. The tree associated with n fields is exhausted when the root node n is processed and no new edit is generated by algorithm 1 during the last traversal of the tree, or the edit $A_1 x, \dots, x A_n$ has been generated.

Algorithm 1.

1. Form the field index ordered tree.
2. Form the "fields that enter" tableau, which consists of a listing of the edits each field enters.
3. Start with root node 1.
4. Form the "field code" matrix F_* associated with the current node $*$. This is a matrix whose entries are 0,1, whose rows correspond to all previously identified edits which have essential fields the antecedent of $*$, and whose columns correspond to the component entries of the field indexed by the last digit of $*$, which shall be called k . Those entries j that are not actively constrained by edit i have corresponding entry $a_{ij} = 0$. Otherwise, the entry is 1. (The antecedent of a node $** \dots **$ is the node $** \dots *$, and the antecedent of a root node is empty insofar as the "field code" matrix is formed from the original explicit set of edits.)
5. For the "field code" matrix F_* , find all minimal solutions (primal coverings) to the problem

$$\begin{aligned} wF_* &\geq 1 \\ w_i &= 0,1 \end{aligned}$$

6. For each minimal solution (primal covering) w to step 5, use field k as a generating field in the Fellegi-Holt construction (lemma 2) with each set S equal to the indices associated with the nonzero components of the respective minimal solution w . Discard all generated edits which are either empty or included in some explicit or previously generated implied edit.
7. Update the "fields that enter" tableau by eliminating those edits completely contained in a newly generated edit and by adding the newly generated edit.

8. If the tree is not exhausted, then proceed to step 4 with the next node (as per the ordering defined by the field index ordered tree). (The next root node after n is defined to be root node 1.)
9. If the tree is exhausted, form the edit matrix from the remaining explicit edits and nondiscarded implied edits. Stop.

Example 5

Let the data space be $A_1 \times \dots \times A_6$, where $A_1 = \{0,1\}$, $A_2 = \{0,1,2\}$, $A_3 = \{0,1\}$, $A_4 = \{0,1,2,3\}$, $A_5 = \{0,1,2\}$, $A_6 = \{0,1,2,3\}$.

Let the edits e_1 - e_5 be defined as below:

$$\begin{aligned}
 e_1 &= A_1 \times \{0,1\} \times \{0\} \times A_4 \times \{0,1\} \times A_6 \\
 e_2 &= \{1\} \times A_2 \times \{1\} \times \{0,1\} \times A_5 \times \{2,3\} \\
 e_3 &= \{0\} \times \{1,2\} \times A_3 \times \{1,2,3\} \times A_5 \times A_6 \\
 e_4 &= A_1 \times \{0,2\} \times A_3 \times A_4 \times A_5 \times \{0,1\} \\
 e_5 &= \{1\} \times A_2 \times A_3 \times \{0\} \times \{1,2\} \times A_6
 \end{aligned}$$

The fields that enter tableau are:

1	2	3	4	5	6
e_2	e_1	e_1	e_2	e_1	e_2
e_3	e_3	e_2	e_3	e_5	e_4
e_5	e_4		e_5		

The first "field code" matrix is associated with root node 1, and is given by

$$F_1 = \begin{vmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{vmatrix}.$$

There are two primal coverings e_2, e_3 and e_3, e_5 . Edits e_2 and e_3 with generating field 1 yield the implied edit $e_6 = A_1 \times \{1,2\} \times \{1\} \times \{1\} \times A_5 \times \{2,3\}$. Edits e_3 and e_5 with generating field 1 yield the empty implied edit $A_1 \times \{1,2\} \times A_3 \times \phi \times \{1,2\} \times A_6$.

The implied edits generated are listed in sequence by location of generation.

Node	First tree traversal	Generating edits
1	$e_6 = A_1 \times \{1,2\} \times \{1\} \times \{1\} \times A_5 \times \{2,3\}$	$e_2 \quad e_3$
12	$e_7 = A_1 \times A_2 \times \{0\} \times A_4 \times \{0,1\} \times \{0,1\}$	$e_1 \quad e_4$
13	$e_8 = A_1 \times \{1\} \times A_3 \times \{1\} \times \{0,1\} \times \{2,3\}$	$e_1 \quad e_6$
16	$e_9 = A_1 \times \{2\} \times \{1\} \times \{1\} \times A_5 \times A_6$	$e_4 \quad e_6$
2	$e_{10} = \{0\} \times A_2 \times \{0\} \times \{1,2,3\} \times \{0,1\} \times A_6$	$e_1 \quad e_3$
2	$e_{11} = \{0\} \times A_2 \times A_3 \times \{1,2,3\} \times A_5 \times \{0,1\}$	$e_3 \quad e_4$
25	$e_{12} = \{1\} \times A_2 \times \{0\} \times \{0\} \times A_5 \times \{0,1\}$	$e_5 \quad e_7$
3	$e_{13} = \{1\} \times \{0,1\} \times A_3 \times \{0,1\} \times \{0,1\} \times \{2,3\}$	$e_1 \quad e_2$
35	$e_{14} = \{1\} \times \{0,1\} \times A_3 \times \{0\} \times A_5 \times \{2,3\}$	$e_5 \quad e_{13}$
5	$e_{15} = \{1\} \times \{0,1\} \times \{0\} \times \{0\} \times A_5 \times A_6$	$e_1 \quad e_5$
56	$e_{16} = \{1\} \times \{0,2\} \times \{1\} \times \{0,1\} \times A_5 \times A_6$	$e_2 \quad e_4$
56	$e_{17} = \{1\} \times \{0\} \times A_3 \times \{0\} \times A_5 \times A_6$	$e_4 \quad e_{14}$
6	$e_{18} = \{1\} \times \{0\} \times A_3 \times \{0,1\} \times \{0,1\} \times A_6$	$e_4 \quad e_{13}$

4.1 Second Tree Traversal — No New Edits

In light of the emphasis on efficiency, it is immediate that algorithm 1 is barely acceptable; it requires iterated traversal of the complete tree and often results in rather large set covering problems at step 5. For example, at root node 3, the field code matrix F_3 is

e_1	1	0
e_2	0	1
e_6	0	1
e_7	1	0
e_9	0	1
e_{10}	1	0
e_{12}	1	0

and hence yields a multiplicity of solutions (primal coverings). However, it develops that only the solution $\{e_1, e_2\}$ yields a new implied edit, edit e_{13} .

Fortunately, algorithm 1 can be trimmed considerably, and theorems 2 and 3 provide the basis for this trimming.

Theorem 2 (Commutativity). Let $(i_1, \dots, i_r)$ be a subset of the field indices $(1, \dots, n)$ and let σ be a permutation of the indices which maps the subset $(i_1, \dots, i_r)$ into itself. Furthermore, for any set S of edits and any field $i \in (1, \dots, n)$, let $f_i(S)$ be another set of edits defined by $e \in f_i(S)$ if and only if either e is an edit of S with essential field i , or field i is an essential field of e and e is generated from edits of S by the mechanism of lemma 2 with field i the generating field. Now let E be the set of explicit edits and $E_{i_1} = f_{i_1}(E)$. Define $E(i_1, \dots, i_k) = f_{i_k}(E(i_1, \dots, i_{k-1}))$. Then $E(i_1, \dots, i_r) = E_{\sigma(i_1, \dots, i_r)}$.

Proof. See appendix.

Theorem 3. Define a set S of edits to be locally redundant in the generation of essential edits with essential fields $i_1, \dots, i_j$ from edits of $\hat{E}$ if and only if for every point x of the data space satisfying $x \in e$ with $e \in f_{i_j} \circ \dots \circ f_{i_1}(\hat{E}, S)$, there exists $\hat{e} \in f_{i_j} \circ \dots \circ f_{i_1}(\hat{E})$ also satisfying $x \in \hat{e}$. Consider two ordered sets of indices A and B and generated edits E_A and E_B (as per theorem 3). Then for any index set B , the set of edits E_B is locally redundant in the generation of E_A .

Proof. See appendix.

Corollary 1. If $x \in e$, with $e \in f_{i_j} \circ \dots \circ f_{i_1}(\hat{E}, S)$ and e has essential fields $i_1, \dots, i_j, i_{j+1}, \dots, i_k$, then there exists $\hat{e} \in f_{i_k} \circ \dots \circ f_{i_1}(\hat{E})$ satisfying $x \in \hat{e}$.

Proof. It is immediate that $e \in f_{i_k} 0, \dots, 0 f_{i_{j+1}} 0 f_{i_j} 0, \dots, 0 f_{i_1} (\hat{E}, S)$. Now use theorems 2 and 3.

The consequence of these two theorems is that first, the tree associated with algorithm 1 need only be traversed once; second, that in the formulation of the respective field code matrices, one needs only the explicit edits and the edits generated at the antecedent nodes; and third, if no new edit is generated at the present node, the next node considered is the node for which the present node is not an antecedent. For example, in the case of example 4, if node {1,2,3} generates no new edits, then node {1,3} is the next node considered. Moreover, the order of traversal is somewhat flexible. It remains to prove that this modification of algorithm 1 does indeed generate a sufficient collection of edits.

Theorem 4. Let E be a set of explicit edits. Generate implied edits according to algorithm 1, with a single traversal of the field index ordered tree, and with only the explicit edits and edits generated at the antecedent nodes (or their replacements — step 6 of algorithm 1) considered in the formulation of the respective field code matrices, and such that if no new edit is generated at a node, the next node considered is the next node for which the present node is not an antecedent. Then the collection of remaining explicit and generated implied edits is a sufficient collection of edits.

Proof. By lemma 3, for any edit e , an edit e^* satisfying $e \subset e^*$ can be generated by repeated application of lemma 2. The proof at theorem 4 proceeds by induction on k , the total number of applications of lemma 2 to generate e^* . Formally, k is defined recursively: If the last application of lemma 2 involved edits e^r , $r \in S$, then $k = n(e^*) = \sum_{r \in S} n(e^r)$. Let e^* be an edit satisfying $e \subset e^*$ generated by a minimal number of applications of lemma 2.

1. $k = 1$. In this case all of the edits used in the application of lemma 2 are explicit edits. Without loss of generality, let the generating field be field 1. If field 1 is an essential field of e^* , then by minimality it is essential in none of the explicit edits used to generate e^* , and e^* can be generated according to the specifications of theorem 4. Otherwise, let $x = (x_1, \dots, x_n)$ be a fixed, but arbitrary,

point of e^* . By definition, there is an explicit edit containing x , with essential fields at least those of e^* , and hence of e .

2. $k = t - 1$. Assume the result, that is, if e is an edit and e^* is an edit generated by $t - 1$ repeated applications of lemma 2 and $x \in e^*$, then there is an edit $\hat{e}$ generated according to the specifications of theorem 4 with essential fields at least those of e^* and satisfying $x \in \hat{e}$.

3. $k = t$. In this case, there is no edit e^* satisfying $e \subset e^*$ that can be generated in fewer than t applications of lemma 2. Fix some edit e^* generable by t applications of lemma 2, and fix the order of applications of the lemma. Without loss of generality, let the generating field at the final step be field 1. If field 1 is not essential, then the proof is complete by the induction hypothesis. Otherwise, let $x = (x_1, \dots, x_n) \in e^*$. By construction, for each value $a_i \in A_1$, there exists an edit e_i involved in the generation of e^* , with essential fields (other than field 1) at least those of e^* , and with the property that $(a_i, x_2, \dots, x_n) \in e_i$. Each edit e_i can be generated by no more than $t - 1$ applications of lemma 2, and hence by hypothesis, there exist edits $\hat{e}_i$ which can be generated according to theorem 4, which have the same essential fields as each of the e_i (respectively), and satisfy $(a_i, x_2, \dots, x_n) \in \hat{e}_i$. Let the essential fields of e^* be 1, ..., m . According to the notation of theorem 2, there exist index sets B_i such that $\hat{e}_i \in E_{B_i}$ and $x \in f_1(E_{B_1}, E_{B_2}, \dots)$. It follows that $x \in f_1 \circ \dots \circ f_m(E_{B_1}, E_{B_2}, \dots) \subset f_1 \circ \dots \circ f_m(E, E_{B_1}, E_{B_2}, \dots)$, and hence by theorems 2 and 3, that $x \in f_m \circ \dots \circ f_1(E)$. All that remains to complete the proof is to show that the "truncation property" of theorem 4 can be observed. The relevant issue is that perhaps at some stage prior to the application of the operator f_m , no new edit is generated. In this case, begin with the operator f_1 and continue until the first occurrence that no new edit is generated; let it be at stage f_i . Omit the operator f_i and continue "pruning" operators until the complete sequence is processed. Let the resulting sequence be $f_{i_k} \circ \dots \circ f_{i_1}$. It is easy to see that $f_m \circ \dots \circ f_1(E) = f_{i_k} \circ \dots \circ f_{i_1}(E)$.

Additional heuristics are also available; sometimes to more quickly be able to determine when a generated edit is empty, the edit can be grouped by fields and orthogonality classes.

For example 5, the orthogonality classes are as follows:

	F_1	F_2	F_3	F_4	F_5	F_6
Ess	e_1, e_4	e_2, e_5	e_3, e_4, e_5	e_1, e_4	e_2, e_3, e_4	e_1, e_3, e_5
Class 1	e_2, e_5	e_1, e_3		e_2, e_3	e_1, e_5	
Class 2		e_1, e_4		e_2, e_5		
Class 3		e_3, e_4				

(A collection C of edits are placed in the same location of the orthogonality class matrix if and only if for a given field either (1) the field is essential for each of the edits in the collection C — the (Ess, F_i) entries, or (2) the field is not essential and there is no collection properly containing C such that all the edits of the collection contain a common field element. Thus edits e_1 and e_3 both contain the element 1 in field 2, but edits e_1, e_3 , and e_4 have no element in common.)

This table allows for quick checking of whether a collection of edits will generate a nonempty implied edit; for other than the generating field, one checks whether all elements belong either to the same class or are essential in that field. For example, edits e_3 and e_5 with generating field 1 do not satisfy either of the above conditions for field 4 and hence cannot generate a nonempty edit. (These heuristics can be formalized, and will be if additional computational experience shows that they appear to be significant.)

Selected steps of the generation of the implied edits of example 5 are given below. (It should be noted, however, that the generation sequence was not in strict adherence to the field index tree.)

Essential edits

Fields that enter tableau

E_*

	1	2	3	4	5	6
e_2		e_1	e_1	e_2	e_1	e_2
e_3		e_3	e_2	e_3	e_5	e_4
e_5		e_4		e_5		

Class	Essential edits	Generating field	Edits used to generate
E_1	e_1	—	—
	e_4	—	—
	e_6	1	e_2 and e_3
E_2	e_2	—	—
	e_5	—	—
	e_7	2	e_1 and e_4
	e_{10}	2	e_1 and e_3
	e_{11}	2	e_3 and e_4
E_3	e_3	—	—
	e_4	—	—
	e_5	—	—
	e_{13}	3	e_1 and e_2
E_4	e_1	—	—
	e_4	—	—
E_5	e_2	—	—
	e_3	—	—
	e_4	—	—
	e_{15}	5	e_1 and e_5
E_6	e_1	—	—
	e_3	—	—
	e_5	—	—
	e_{16}	6	e_2 and e_4

Fields that enter tableau

Essential edits

 $E_{1,*}$

2	3	4	5	6
e_1	e_1	e_6	e_1	e_4
e_4	e_6			e_6
e_6				

Class	Essential edits	Generating field	Edits used to generate
$E_{1,3}$	e_4	—	—
	e_8	3	e_1 and e_6
$E_{1,6}$	e_1	—	—
	e_9	6	e_4 and e_6

Fields that enter tableau

Essential edits

 $E_{2,*}$

	3	4	5	6
	e_2	e_2	e_5	e_2
	e_7	e_5	e_7	e_7
	e_{10}	e_{10}	e_{10}	e_{11}
		e_{11}		

Class	Essential edits	Generating field	Edits used to generate
$E_{2,5}$	e_2 e_{12}	— 5	— e_5 and e_7

Fields that enter tableau

Essential edits

 $E_{3,*}$

	4	5	6
	e_3	e_5	e_4
	e_5	e_{13}	e_{13}
	e_{13}		

Class	Essential edits	Generating field	Edits used to generate
$E_{3,5}$	e_3 e_{11} e_{14}	— — 5	— — e_5 and e_{13}
$E_{3,6}$	e_3 e_5 e_{18}	— — 6	— — e_4 and e_{13}

Fields that enter tableau

Essential edits

 $E_{3,5,*}$

	6
	e_4
	e_{14}

Class	Essential edits	Generating field	Edits used to generate
$E_{3,5,6}$	e_3 e_{17}	— 6	— e_4 and e_{14}

Heretofore, no explicit mention has been made as to how best to deal with edits that restrict only one field, that is, with only one field entering (In terms of definition 1, for any field i , these are the edits that specify the feasible values A_i). A unified approach is easily suggested:

Assume that a record x fails single field edits $e_1, \dots, e_j$, with fields that enter $1, \dots, j$ respectively. Then it is clear that the fields $1, \dots, j$ must be included in any solution to the error localization problem, that is

$$x + y \in \text{acceptance region} \Rightarrow y_i \neq 0 \text{ for } i = 1, \dots, j.$$

Now let $e_{j+1}, \dots, e_k$ be the collection of edits for which fields $1, \dots, j$ are all essential and which the record x fails. Then the solution of the fields to impute problem is simply the solution to the set-covering problem

$$\min \sum c_i w_i$$

subject to

$$\begin{aligned} Mw &\geq 1 & w_i &= 0, 1 \\ & & w_i &= 1 \text{ for } i = 1, \dots, j \end{aligned}$$

and M is the failed-edit matrix determined by edits $e_{j+1}, \dots, e_k$.

5. NORMAL EDIT GENERATING ALGORITHM

The type of edit format that this algorithm is least able to handle is a collection of many edits, each sparsely represented so that the "fields that enter" tableau has many entries. In any case, the more numerous the explicit edits, the more cumbersome the Boolean approach. For this reason and for the reason that originally each of the edits may not be given in normal form by the experts in the subject matter, an additional algorithm is proposed. This algorithm reexpresses any locus

of points in a lattice as a union of normal forms and does so in a minimal fashion, insofar as no subset of the generated normal edits covers all of the edit points.

Algorithm 2. An algorithm for normal form reexpression of edits.

1. Order the coordinates.
2. Order and list all edit points. Call this list A and replicate as list B.
3. Begin with the first point in list A and in the first coordinate direction form the maximal "connected" interval containing the first point (list B is used for this purpose). Strike from list A those points so identified.
4. Proceed in the direction of the second coordinate and determine the maximum connected rectangle containing the interval determined in step 3. Again, strike from list A those points so identified.
5. For each $j < n$, extend the j dimensional generalized rectangle to a $j + 1$ dimensional one in the $j + 1$ coordinate direction.
6. When $j = n$, if list A is empty, stop. Otherwise, proceed with step 3 with the next point not stricken from list A.

Example 6 (screening question)

Assume a sequence of three questions with possible answers as below:

1. Yes	No	
2. Yes	No	
3. A	B	C

Furthermore, assume that question 3 must be answered only if the first two questions are answered "yes", and that any combination of "no" answers requires that question 3 be unanswered.

The coding can be done as follows:

Codification of example 5

Questions 1 and 2:	Yes	→ 0
	No	→ 1
	No answer	→ 2
Question 3:	A	→ 0
	B	→ 1
	C	→ 2
	No answer	→ 3

The acceptance region is $\{0\} \times \{0\} \times \{0,1,2\}$, $\{0,1,2\} \times \{1\} \times \{3\}$, $\{1\} \times \{0,1,2\} \times \{3\}$, and the failure region is the complement of the acceptance region.

The failure region can be expressed as 28 (point) edits.

Edit points for example 5

(0,0,3)	(0,2,3)	(1,2,0)	(2,1,0)
(0,1,0)	(1,0,0)	(1,2,1)	(2,1,1)
(0,1,1)	(1,0,1)	(1,2,2)	(2,1,2)
(0,1,2)	(1,0,2)	(2,0,0)	(2,2,0)
(0,2,0)	(1,1,0)	(2,0,1)	(2,2,1)
(0,2,1)	(1,1,1)	(2,0,2)	(2,2,2)
(0,2,2)	(1,1,2)	(2,0,3)	(2,2,3)

The algorithm first picks off the point (0,0,3) and identifies it as a singleton point. It then passes to the point (0,1,0) and identifies the interval (0,1,0); (1,1,0); (2,1,0) or in binary form:

Field 1			Field 2			Field 3			
0	1	2	0	1	2	0	1	2	3
1	1	1	*	1	*	1	*	*	*

Next, it determines that a similar interval with second coordinate equal to 0 does not exist, but that one with second coordinate equal to 2 does. Hence, it yields the rectangle:

Field 1			Field 2			Field 3			
0	1	2	0	1	2	0	1	2	3
1	1	1	0	1	1	1	*	*	*

*Indeterminate at this stage.

Finally, it identifies the solid:

Field 1			Field 2			Field 3			
0	1	2	0	1	2	0	1	2	3
1	1	1	0	1	1	1	1	1	0

Proceeding in this fashion, the algorithm reexpresses the 28 edits as 6 edits as in the diagram below. The numbers give the order in which the edits are generated.

Six edits of example 5 in edit matrix format

	Field 1			Field 2			Field 3			
	Y	N	No Ans	Y	N	No Ans	A	B	C	No Ans
e ₁	1	0	0	1	0	0	0	0	0	1
e ₂	1	1	1	0	1	1	1	1	1	0
e ₃	1	0	0	0	0	1	1	1	1	1
e ₄	0	1	1	1	1	1	1	1	1	0
e ₅	0	0	1	1	0	0	1	1	1	1
e ₆	0	0	1	0	0	1	1	1	1	1

Noteworthy is the fact that in this particular case, subsequent application of algorithm 1 generates no new essential implied edits. Unfortunately, such is not always the case, and in fact, often algorithm 2 yields no gains.

6. CONCLUSIONS

The principal result of this paper is the formulation of the concept of a sufficient set of edits and the development of an improved algorithm to generate such a set; the justification of the algorithm appears in theorems 2, 3, and 4. It is hoped that this algorithm will substantially expedite the localization of errors. Associated results are an algorithm

to express an edit set as a union of normal form edits and the reduction of localization of errors to a set-covering problem.

Additional refinements are known, in special cases, which further expedite the localization of errors (renumbering the components within a field, certain row and column domination methods applicable to the resultant set-covering problem), yet these, as well as the algorithm presented here, await additional computational experience before a final judgment is issued with regard to their general usefulness.

ORNL-DWG 79-13465

Fig. 1. Graphical illustration of edit points of example 6.
Inefficient expression: 28 edits.

ORNL - DWG 79 - 19375

Fig. 2. Diagram of edits of example 6 after application of algorithm 2.

ORNL-DWG 79-13466

Fig. 3. Exploded diagram of edits of example 6 after application of algorithm 2.

APPENDIX

Theorem 2 (Commutativity). Let $(i_1, \dots, i_r)$ be a subset of the field indices $(1, \dots, n)$ and let σ be a permutation of the indices which maps the subset $(i_1, \dots, i_r)$ onto itself. Furthermore, for any set S of edits and any field $i \in (1, \dots, n)$, let $f_i(S)$ be another set of edits defined by $e \in f_i(S)$ if and only if either e is an edit of S with essential field i , or field i is an essential field of e and e is generated from edits of S by the mechanism of theorem 2 with field i the generating field. Now let E be the set of explicit edits and $E_{i_1} = f_{i_1}(E)$. Define $E_{(i_1, \dots, i_k)} = f_{i_k}(E_{(i_1, \dots, i_{k-1})})$. Then $E_{(i_1, \dots, i_r)} = E_{\sigma(i_1, \dots, i_r)}$.

Proof of theorem 2. Without loss of generality, let $(i_1, \dots, i_r) = (1, \dots, r)$. For each of the edits of either $E_{(1, \dots, r)}$ or $E_{\sigma(1, \dots, r)}$, any given field can be written as the repeated intersections and union of the corresponding field of certain of the explicit edits, and those fields indexed by $k > r$ can be expressed strictly as intersections. Hence, for $k > r$, the k^{th} field e^k of an edit $e \in E_{(1, \dots, r)}$ can be written as

$$e^k = \bigcap_{S_r} [\bigcap_{S_2, \dots, r} [\bigcap_{S_1, \dots, r} e^k_{i_1, \dots, i_r}] \dots] ,$$

$$e_{i_1, \dots, i_r} \in E$$

and for $k \leq r$, the k^{th} field e^k of the edit e can be written as

$$e^k = \bigcap_{S_r} [\bigcap_{S_{k+1}, \dots, r} [\bigcup_{S_k, \dots, r} [\bigcap_{S_{k-1}, \dots, r} \dots]$$

$$[\bigcap_{S_1, \dots, r} e^k_{i_1, \dots, i_r}] \dots] ,$$

$$e_{i_1, \dots, i_r} \in E$$

In both of these equations, $e_{i_1, \dots, i_r}^k$ is the k^{th} field of some explicit edit of E and the indexing can clearly be redundant in the sense that it can happen that $(i_1, \dots, i_r) \neq (j_1, \dots, j_r)$, but that $e_{i_1, \dots, i_r} = e_{j_1, \dots, j_r}$. By using shorthand, and with some abuse of notation, the edit e can be written as

$$\left(\bigcap_{S_r} \dots \bigcup_{S_1}, \bigcap_{S_r} \dots \bigcup_{S_2} \bigcap_{S_1}, \dots, \bigcup_{S_r} \bigcap_{S_{r-1}} \dots \bigcap_{S_1}, \right. \\ \left. \bigcap_{S_r} \dots \bigcap_{S_1}, \dots \right) (e_{i_1, \dots, i_r}) \quad (1)$$

Note: In the expression for e , the index set $S_j, \dots, r$ has been abbreviated as S_j . An element e^σ of $E_{\sigma(1), \dots, r}$ can be written similarly except that the order of the unions and intersections in the first r fields are permuted by σ , and the explicit edits $\{\hat{e}_{i_1, \dots, i_r}\}$ may not be the same as edits $\{e_{i_1, \dots, i_r}\}$. For example, the first r field of an element $e^\sigma \in E_{\sigma(1), \dots, r}$ can be written as

$$\bigcap_{S_r^\sigma} \dots \bigcap_{S_{\sigma(1)+1}^\sigma} \bigcup_{S_{\sigma(1)}^\sigma} \dots \bigcap_{S_1^\sigma} \hat{e}_{i_1, \dots, i_r} :$$

By de Morgan's laws, the order of the intersections and union can be changed. In fact, the permutation σ^{-1} applied coordinate-wise simultaneously to all fields of the original representation of e^σ results in a representation of the form (1), but with index sets $\{\bar{S}_j^\sigma, \dots, r\}$ defined as rather complicated functions of the fields and the index sets $\{S_r^\sigma\}, \dots, \{S_1^\sigma, \dots, r\}$. Hence, the edit e^σ can also be written as

$$\begin{aligned}
 & \left(\bigcap_{\sigma} \dots \bigcup_{\sigma} \frac{\sigma}{S_1}, \bigcap_{\sigma} \dots \bigcup_{\sigma} \frac{\sigma}{S_2} \bigcap_{\sigma} \frac{\sigma}{S_1}, \dots, \bigcup_{\sigma} \bigcap_{\sigma} \frac{\sigma}{S_{r-1}} \bigcap_{\sigma} \frac{\sigma}{S_1}, \bigcap_{\sigma} \bigcap_{\sigma} \frac{\sigma}{S_1}, \dots \right) \\
 & (\hat{e}_{i_1}, \dots, i_r) \quad (2)
 \end{aligned}$$

For each element $e^\sigma \in E_\sigma(1, \dots, r)$, each j satisfying $1 \leq j \leq r-1$, and each fixed index set $\zeta_j = (i_{j+1}^0, \dots, i_r^0)$ of $S_{j+1}^\sigma, \dots, r$ define the element e_{j, ζ_j}^σ as the edit each of whose fields is determined by the first $r-j+1$ Boolean expressions of expression (2). That is, $e_{j, \zeta_j}^\sigma =$

$$\begin{aligned}
 & \left(\bigcap_{\sigma} \dots \bigcup_{\sigma} \frac{\sigma}{S_1}, \bigcap_{\sigma} \dots \bigcup_{\sigma} \frac{\sigma}{S_2} \bigcap_{\sigma} \frac{\sigma}{S_1}, \dots, \bigcup_{\sigma} \bigcap_{\sigma} \frac{\sigma}{S_{j-1}} \dots \bigcap_{\sigma} \frac{\sigma}{S_1}, \dots \right) \\
 & (\hat{e}_{i_1}, \dots, i_r) \quad ,
 \end{aligned}$$

with the restriction that $(i_{j+1}, \dots, i_r) = (i_{j+1}^0, \dots, i_r^0)$. If

$$\bigcup_{\sigma} \bigcap_{\sigma} \dots \bigcap_{\sigma} \hat{e}_{i_1}, \dots, i_r = A_{j+1}$$

whenever $(i_{j+2}^0, \dots, i_r^0) \in S_{j+2}^\sigma, \dots, r$, and A_j is the range of feasible values for the j^{th} field, then for $j+1$ and $(i_{j+2}^0, \dots, i_r^0) = \zeta_{j+1}$, it follows that

$$e_{j+1, \zeta_{j+1}}^\sigma = f_{j+1} \{e_{j, \zeta_j}^\sigma \mid \zeta_j = (i_{j+1}^0, i_{j+2}^0, \dots, i_r^0)\}$$

and $\zeta_j \in S_{j+1}^\sigma, \dots, r$.

Hence, by induction, to show that the implied edit $\phi \neq e^\sigma \in E_\sigma(1, \dots, r)$ is also an element of $e(1, \dots, r)$, it suffices to show that

$$\bigcup_{\sigma} \bigcap_{\sigma} \dots \bigcap_{\sigma} \hat{e}_{i_1}^{j+1}, \dots, i_r = A_{j+1}$$

whenever $(i_{j+2}^0, \dots, i_r^0) \in S_{j+2}^\sigma, \dots, r$.

By assumption, the element $e^\sigma \in E_{\sigma}(1, \dots, r)$ and, hence, each of the first r fields of the expression (2) is essential. Moreover, for the j^{th} field, all of the Boolean operators in the expression (2) after the j^{th} are intersections. Thus, the required condition is verified.

Theorem 3. Define a set S of edits to be locally redundant in the generation of essential edits with essential fields $i_1, \dots, i_j$ from edits $e \in \hat{E}$ if and only if for every point x of the data space satisfying $x \in e$ with $e \in f_{i_j} \circ \dots \circ f_{i_1}(\hat{E}, S)$, there exists $\hat{e} \in f_{i_j} \circ \dots \circ f_{i_1}(\hat{E})$ also satisfying $x \in \hat{e}$. Consider two ordered sets of indices A and B and generated edits E_A and E_B (as per theorem 3). Then for any index set B , the set of edits E_B is locally redundant in the generation of E_A .

Proof of theorem 3. Let E be the set of explicit edits. Let A and B be sets of ordered indices, and without loss of generality let $A = (1, \dots, r)$ and $f_A = f_r \circ \dots \circ f_1$ and similarly for f_B . It is not hard to see that f_A is idempotent, that is, that $f_A \circ f_A = f_A$ and that for $A' \subset A$, $f_{A'} \circ f_A = f_A \circ f_{A'} = f_{A'}$. Let $A \cap \sim B = \{i \mid i \in A, i \notin B\}$. It follows that $f_A(E, E_B) = f_{A \cap \sim B} \circ f_{A \cap B} [E, f_{A \cap B}(E_B \cap \sim A)] = f_{A \cap \sim B}(E_B \cap A, E_B)$.

Thus, the theorem need only be proved for the case that $A \cap B = \emptyset$.

Step 1. If $x \in e \in f_1(E, E_B)$, then there exists $\hat{e} \in f_1(E)$ such that $x \in \hat{e}$.

Let $E_B = \{g_i\}$. Then any element of $f_1(E, E_B)$ can be written as

$$\bigcup_{S_1} e_i^1 \cup \bigcup_{S_2} g_i^1, \bigcap_{S_1} e_i^2 \cap \bigcap_{S_2} g_i^2, \dots, \quad (3)$$

where for $j \in B$, $g_i^j = A_j$, the j^{th} field, and

$$\bigcup_{S_1} e_i^1 \cup \bigcup_{S_2} g_i^1 = A_1.$$

Let those $e_i \in E$ which are used in the expression of any $g_i \in S_2$ be indexed by $1, \dots, k$. Then

$$\bigcup_{S_1} e_i^1 \cup e_j^1 = A_1$$

for $j = 1, \dots, k$ and any point of the expression (3) is contained in the union (2)

$$\bigcup_{j=1, \dots, k} [\bigcup_{S_1} e_i^1 \cup e_j^1, \bigcap_{S_1} e_i^2 \cap e_j^2, \dots] \quad (4)$$

Step 2. If $x \in e \in f_2$ or $f_1(E, E_0)$, then there exists $\hat{e} \in f_2$ or $f_1(E)$ such that $x \in \hat{e}$, and hence, by induction, the proof is complete.

For each value a_i of A_2 , choose an element $e_i \in E_1$ covering $(x_1, a_i, x_3, \dots, x_n)$, that is, an element e_i satisfying $(A_1, a_i, x_3, \dots, x_n) \subset e_i$ (existence assured by step 1). Then it follows that

$$x \in (\cap e_i^1, \cup e_i^2, \cup e_i^3, \dots) = (A_1, A_2, \cap e_i^3, \dots) .$$

REFERENCES

1. I. P. Fellegi and D. Holt, "A Systematic Approach to Automatic Edit and Imputation," *J. Am. Stat. Assoc.* (353): 17-35 (1976).
2. R. S. Garfinkel, "An Algorithm for Optimal Imputation of Erroneous Data," in preparation (1979).
3. R. S. Garfinkel and G. L. Nemhauser, *Integer Programming*, Wiley-Interscience, New York (1972).
4. G. E. Liepins, "A Rigorous, Systematic Approach to Automatic Data Editing and Its Statistical Basis," in preparation (1979).

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

INTERNAL DISTRIBUTION

1. J. Arehart	22. D. C. Parzyck
2. B. H. Bronfman	23. J. W. Sims
3. R. L. Burgess	24. C. L. Sullivan
4. R. S. Carlsmith	25. V. R. R. Uppuluri
5. M. R. Chernick	26. C. Weisbin
6. R. M. Davis	27. G. Westley
7. P. Dizillo-Benoit	28. T. Wilbanks
8. W. Fulkerson	29. T. Wright
9. D. A. Gardiner	30-31. Central Research Library
10. R. Honea	32. Document Reference Section
11. W. E. Lever	33-34. Laboratory Records
12-21. A. S. Loeb1	35. Laboratory Records - RC
	36. ORNL Patent Office

EXTERNAL DISTRIBUTION

37. Prof. Jean Abadie, Institut de Programmation, Université de Paris, VI Place Jussieu, Paris, France
38. Barbara Bailar, Associate Director for Statistical Standards and Methodology Bureau of the Census, Room 2031, Federal Building 3, Washington D.C. 20233
39. Richard T. Beckman, S-1 Statistics, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
40. M. Brockway, Energy Resources Center, The University of Oklahoma, P. O. Box 3030, Norman, OK 73070
41. L. Buzbee, C-3, Applications Support and Research, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
42. W. J. Conover, Chairman, Department of Statistics Business School, Texas Technological University, Lubbock, TX 79409
43. L. Cox, Statistical Research Division, Bureau of Census, Suitland, MD 20023
44. Dave Croot, Assistant Director of Projects, System Development Division, 14-E, R. H. Coates Building, Statistics Canada, Tunney's Pasture, Ottawa, Ontario, Canada KIA OT6
45. D. J. DePriest, Office of Naval Research, 800 North Quincy Street, Arlington, VA 22217
46. R. G. Easterling, Division 1223, Sandia Laboratories, 7800 Northridge NE, Albuquerque, NM 87109
47. I. P. Fellegi, Statistics Canada, Tunney's Pasture, Ottawa, Ontario, Canada KIA OT6
48. R. Garfinkel, Management Science Department, University of Tennessee, Knoxville, TN 37916
49. T. L. Gilbert, EIS-10, Argonne National Laboratory, Argonne, IL 60439

50. S. Grady, Science Applications, Inc., 100 Jackson Plaza, Oak Ridge, TN 37830
51. Ron Graves, 12-0 Coates Tower, Statistics Canada, Tunney's Pasture, Ottawa, Ontario, Canada KIA OT6
52. B. Greenberg, Statistical Research Division, Bureau of Census, Suitland, MD 20023
53. H. J. Greenberg, Energy Information Administration, U.S. Department of Energy, 12th and Pennsylvania Avenues, N.W., Washington, D.C. 20461
54. H. O. Hartley, 1804 Woodburn, Durham, NC 27705
55. R. E. Huddleston, Applied Mathematics Division 8332, Sandia Laboratories, Livermore, CA 94550
56. R. L. Iman, Division 1223, Sandia Laboratories, Albuquerque, NM 87185
57. S. C. Kao, Applied Math Department, Brookhaven National Laboratory, Upton, NY 11973
58. R. J. Kee, Applied Math Division 8331, Sandia Labs, Livermore, CA 94550
59. R. K. Lohrding, S-1, Statistics, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
60. V. Lowe, Lawrence Livermore Laboratory, P.O. Box 808, Livermore, CA 94550
61. G. Lüders, The Analytic Sciences Corp., Six Jacob Way, Reading, MA 01867
62. John Mathur, Regional Assessments Division, Department of Energy, EV-242, Mail Stop E-201, Rm D-432, Washington, DC 20545
63. P. McKeown, Quantitative Business Analysis, Univ. of Georgia, Athens, GA 30602
64. P. C. Messina, Applied Mathematics Division, Argonne National Laboratory, Argonne, IL 60439
65. F. L. Miller, Jr., Desert Research Institute, 4582 Maryland Parkway, Las Vegas, NV 89109
66. J. Miller, Department of Math, Texas Tech. University, Lubbock, TX 79409
67. T. J. Mitchell, Mathematical Research Center, University of Wisconsin, 610 North Walnut St., Madison, WI 53706
68. L. E. Moses, Administrator, Energy Information Administration, U.S. Dept. of Energy, 12th and Pennsylvania Avenues, NW, DC 20461
69. M. Murphy, Energy Resources Center, The University of Oklahoma, P.O. Box 3030, Norman, OK 73070
70. W. L. Nicholson, Pacific Northwest Laboratory, P.O. Box 999, Richland, WA 99352
71. A. Peshkin, Applied Mathematics Department, Brookhaven National Laboratory, Upton, NY 11973
72. J. C. T. Pool, Office of Basic Energy Sciences, ER-17, J-309, GTN, U.S. Department of Energy, DC 20545
73. B. A. Poole, Government Technical Representative, Energy Information Administration, U.S. Department of Energy, 12th and Pennsylvania Avenues, NW, DC 20461
74. C. Quong, Computer Science and Applied Mathematics Department, Lawrence Berkeley Laboratory, Berkeley, CA 94720
75. D. M. Rasmuson, EG&G Idaho, Inc., P.O. Box 1625, Idaho Falls, ID 83401

76. S. Reynolds, Energy Information Administration, Department of Energy, 12th and Pennsylvania Avenues, NW, DC 20461
77. G. Sande, Business Survey Methods, Statistics Canada, Tunney's Pasture, Ottawa, Ontario, Canada KIA 0T6
78. D. L. Shaeffer, BDM Corp., 130 Fairbanks Rd., Oak Ridge, TN 37830
79. N. G. Smiriga, Lawrence Livermore Laboratory, P.O. Box 808, Livermore CA 94550
80. G. Tietjen, Los Alamos Scientific Laboratory, MS 600, Box 1663, Los Alamos, NM 87545
81. D. Victory, Department of Math, Texas Tech. University, Lubbock, TX 79409
82. B. Wendroff, T-7, Mathematical Modeling and Analysis, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
83. N. Woodley, Branch Chief for Systems Analysis, SERI, 1617 Cole Boulevard, Golden, CO 80401
84. Assistant Manager for Energy Research and Development
Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830
- 85-267. Distribution Category UC-32
- 268-475. Regional and Urban Studies Distribution, Energy Division