INFORMATION RETRIEVAL IN FILES DESCRIBED USING SETS

by

James W. Welch
&
J. Wesley Graham

Research Report CS-75-09

Department of Computer Science

University of Waterloo
Waterloo, Ontario, Canada

March 1975

To be presented at the International Workshop
Management of Data: Description, Access and Control
May 14-16, 1975

ABSTRACT

A number of file structures (i.e., inverted files and multilist files) can be modelled as a collection of records whose contents are described using sets. With each of these sets is associated a definition which is stated as a Boolean expression of attributes. The contents of a set logically consists of all records in the file for which the associated definition is true. A common application is to determine all records in the file which satisfy a given query (stated as a Boolean expression of attributes).

We consider this retrieval problem in a general manner. A method is developed whereby the descriptive sets are combined, using the conventional set-theoretic operations, to obtain a superset of the required collection of records. This superset can be shown to minimal, with regard to the given query and the given collection of descriptive sets. For an arbitrary query, some of the descriptive sets may be irrelevant and can be ignored when constructing this superset. We present a method whereby the descriptive sets can be partitioned into classes and we show how some of these partitions need not be considered when constructing the superset.

A general model of the file structure, called Set-Theoretical Descriptive Storage (STDS), is formulated and used to analyze the retrieval problem. The minimal superset, obtained in the basic construction, may not be the best superset to process in order to determine the exact collection of records (complementation, for example, may be a prohibitively expensive operation). Because of these considerations we present a number of methods to construct supersets which are constrained in different ways (i.e., complementation of sets is not allowed).

Keywords: attributes, Boolean functions, Boolean queries, file organization, information retrieval, inverted file, multilist file, retrieval

INFORMATION RETRIEVAL IN FILES DESCRIBED USING SETS

1) Introduction:

In several well-known file systems (i.e., inverted files and multilist files) some of the information is classified using sets. Each of these sets has a definition and contains all records for which this definition is true. A common problem is to select all records for which a Boolean expression of keywords or attributes is true. In this paper we shall formally consider this problem.

The definitions used to generate the descriptive sets are Boolean expressions of attributes. For an arbitrary query, we will demonstrate a number of methods to combine the sets, using the usual set-theoretic operations, to obtain a superset of the required information. One of the methods can be used to construct a superset which is minimal. The alternative methods can be used when other considerations, such as the cost of complementing a set, become important. These methods can be employed in a general file structure, Set-theoretical Descriptive Storage (STDS), which we shall define. Several well-known file systems, such as inverted files and multilist files, are special cases of this general definition.

As a preliminary example, consider an inverted file consisting of information about the employees of a company. Each record in this personnel file would correspond to one employee and would contain the data relevant to that employee. An inverted list is a collection of records (or record addresses) for which some property is true. For example, a list of all employees who work in the engineering department might be maintained. There may be many such lists and the lists may have complex definitions. Queries, or retrieval requests, may be specified as Boolean expressions of attributes, i.e., "employee works in engineering department" AND "employee is a secretary". To locate all records for which such a Boolean expression is true, the appropriate lists may be used. For example, the collection of secretaries who work in the engineering department may be determined as the intersection of the lists with the definitions:

"employee works in engineering department"

"employee is a secretary"

If only one of these two lists existed, then the requested collection of records could be determined by examining those records indicated by the defined list.

The retrieval problem, then, introduces several questions:

(i) Given a query, what sets should be used to determine the records for which the query is true?

(ii) How should the sets in (i) be used to determine the required collection?

In section (2) we present an abstraction of the problem and introduce several concepts. We provide an answer to question (ii) in section (3) and we indicate a solution to question (i) in section (4). We present (section (5)) a general definition of a file structure and indicate how the results of the previous sections are applicable. In section (6) we present further solutions to question (ii). We conclude our remarks in section (7).

2) Definitions and an abstraction of the problem

Formally, we present the following (standard) definitions:

Definition [1]: A Bi-valent Boolean variable is a variable which may assume values of 0 or 1.

Definition [2]: A Boolean expression is defined (recursively) as:

(i) 0, 1, and bi-valent Boolean variables are Boolean expressions.

(ii) If f and g are Boolean expressions, then so are $f \vee g$, $f \wedge g$, $\bar{f}$, and $\bar{g}$.

(iii) Only items (i) and (ii) are Boolean expressions.

We shall use the terms "variable" and "expression" as synonyms for "bi-valent Boolean variables" and "Boolean expression", respectively. The operations "$\wedge$" (AND), "$\vee$" (OR), and complementation are defined in the usual sense. For legibility, we shall write expressions omitting "$\wedge$" operations. For convenience, we assume that these implied operations are to be performed before "OR" operations. Thus, the expression $ab \vee cd$ is understood to be equivalent to $(a \wedge b) \vee (c \wedge d)$.

The records in the file have attributes or properties which we shall treat as variables. We shall assume that every variable may be assigned a value of true (1) or false (0) for every record in the file. Definitions of sets associated with the file and the queries are stated as expressions using these attributes. For an arbitrary query we would like to determine the set of records for which the query is true. This set may be constructed using the set-theoretic operations of union, intersection, and complementation, applied to the sets describing the file. When it is not possible to exactly determine this set, we would like to determine a set which contains the required set. Those records for which the query is false are then eliminated from this superset.

These considerations motivate our definition of an f-cover (functional cover) of a Boolean expression.

Definition [3]: An expression f is equivalent to an expression g if and only if for all valuations of the variables in f and g, f = g.

Definition [4]: An expression g covers an expression f (written f=>g) if and only if g is true for any evaluation of the variables for which f is true.

Definition [5]: For an arbitrary expression q, an f-cover of q with respect to the expressions f(1), f(2), ..., f(n) is an expression C[f(1),f(2),...,f(n)] satisfying q => C.

Thus, an f-cover q is an expression, with the given expressions as variables, which covers q. For example, if

$q = ab \vee c$

$f(1) = ab$

$f(2) = \bar{a}c$

$f(3) = ac$

$f(4) = b$

$f(5) = c$

then,

$f(1) \vee f(2) \vee f(3)$

$f(4)\bar{f}(5) \vee f(2) \vee f(3)$

1

are all f-covers of q with respect to the indicated expressions. As we shall see, an expression of special interest is the minimal f-cover.

<u>Definition [6]</u>: A <u>minimal</u> <u>f-cover</u> C of an expression q, with respect to f(1), f(2), ..., f(n), is an f-cover satisfying C=>C' for all f-covers C' of q with respect to f(1), f(2), ..., f(n).

Intuitively, the minimal f-cover is the "smallest" f-cover. Before considering how to construct a minimal f-cover, we find it convenient to present the following lemmas.

<u>Lemma [1]</u>: If $f_1 => g_1$ and $f_2 => g_2$, then

(i) $f_1 \vee f_2 => g_1 \vee g_2$

(ii) $f_1 \wedge f_2 \Rightarrow g_1 \wedge g_2$

(iii) $\bar{g}_1 \Rightarrow \bar{f}_1$ and $\bar{g}_2 \Rightarrow \bar{f}_2$

lemma [2]: $f_1 \vee f_2 \Rightarrow g$ if and only if $f_1 \Rightarrow g$ and $f_2 \Rightarrow g$.

lemma [3]: $f \Rightarrow g_1 \wedge g_2$ if and only if $f \Rightarrow g_1$ and $f \Rightarrow g_2$.

The proofs of these lemmas follow directly from the definitions and are omitted.

3. Construction of minimal f-covers

For an arbitrary expression q and for a collection of expressions f(1), f(2),...,f(n), let $U=\{w(1),w(2),\ldots,w(k)\}$ be the collection of variables used. Then, q can be written in disjunctive normal form as

$$q = \bigvee_{i=1}^{\ell} u(i,1)u(i,2)\ldots u(i,k) \qquad [1]$$

where $u(i,j)$ $(1\le i\le l, 1\le j\le k)$ is either $w(i)$ or $\bar{w}(i)$. Each of the l terms in [1] may be used to define a valuation V(i) of the variables in U as follows. Let $V(i) = \{v(i,1),\ v(i,2),\ \ldots,v(i,k)\}$ $(1\le i\le l)$ where $v(i,j)=1$ if $u(i,j) = w(j)$ and $v(i,j)=0$ if $u(i,j)=\bar{w}(j)$. For each of the l valuations V(i) $(1\le i\le l)$, each of the n expressions f(j) $(1\le i\le l, 1\le j\le n)$ may be evaluated. Define $h(i,j) = f(j)$ $(1\le i\le l,\ 1\le j\le n)$ if f(j) is true for valuation V(i) and define $h(i,j)=\bar{f}(j)$ otherwise. Then, a minimal f-cover of q is given by

$$C(q) = \bigvee_{i=1}^{\ell} h(i,1)h(i,2)\ldots h(i,n). \qquad [2]$$

Before demonstrating that [2] is a minimal f-cover of q, we shall consider an example of the construction. Let

$q = a \vee b$

$f(1) = a$

$f(2) = b$

$f(3) = c$

Then, we have $U = \{a,b,c\}$ and q is written as

$$q = abc \vee ab\bar{c} \vee a\bar{b}c \vee a\bar{b}\bar{c} \vee \bar{a}bc \vee \bar{a}b\bar{c}$$

and so we determine

$$V(1) = \{1,1,1\}$$

$$V(2) = \{1,1,0\}$$

$$V(3) = \{1,0,1\}$$

$$V(4) = \{1,0,0\}$$

$$V(5) = \{0,1,1\}$$

$$V(6) = \{1,0,1\}$$

and the required f-cover is given by

$$C(q) = f(1)f(2)f(3) \vee f(1)f(2)\bar{f}(3) \vee f(1)\bar{f}(2)f(3)$$
$$f(1)\bar{f}(2)\bar{f}(3) \vee \bar{f}(1)f(2)f(3) \vee f(1)\bar{f}(2)f(3)$$

which simplifies to $C(q) = f(1) \vee f(2)$. The following lemmas (proven in the appendix) justify the above construction.

lemma [4]: $C(q)$ is an f-cover of q with respect to $f(1)$, $f(2)$, ..., $f(n)$.

lemma [5]: $C(q)$ is a minimal f-cover of q.

We can thus construct an f-cover for a given expression with respect to an arbitrary collection of expressions. Referring to the preceding example, it is clear that not all of the expressions in the arbitrary collection need always be considered in the construction. For example, the exclusion of $f(3) = c$ in the example does not change the resulting f-cover. In order to determine which expressions can be excluded from consideration we have developed the concept of irrelevant expressions, which is the subject of the next section.

4. Irrelevant expressions

Intuitively, for a collection of expressions f(1), f(2), ..., f(n), an expression f(i) is irrelevent with respect to an expression q if it contributes nothing to the definition of the minimal f-cover of q. Thus, the expression f(3) is irrelevant for q in the last example. Formally, we define

Definition [7]: Given a collection f(1), f(2), ..., f(n) of expressions, an expression f(i) ($1 \le i \le n$) is *irrelevent* for an expression q if and only if the minimal f-cover of q with respect to f(1),f(2),...,f(n) is equivalent to the minimal f-cover with respect to f(1),...,f(i-1),f(i+1),...,f(n).

It is possible to determine some of the expressions which are irrelevant for a given expression by considering partitions of the collection of given expressions. For an expression f, we define W(f) to be the collection of variables necessary for the definition of f. An equivalence relation Θ is defined as follows:

Definition [8]: For a collection F of expressions, two members f(1),f(k) of F satisfy f(1) Θ f(k) if and only if there exists a collection f(2),f(3),...,f(k-1) of expressions in F such that $W[f(i)] \cap W[f(i+1)] \neq \emptyset$ for $1 \le i < k$.

Θ is easily shown to be symmetric, reflexive, and transitive and so is an equivalence relation. Thus, the given collection F may be partitioned into p disjoint equivalence classes F(1), F(2), ..., F(p). Defining $Z(i) = \bigcup_{f \in F(i)} W[f]$ we can also easily show that Z(1), Z(2),...,Z(p) are disjoint. The following lemma and its corollary indicate why this partitioning process is important.

Lemma [6]: Let C be a minimal f-cover for an expression q with respect to f(1), f(2), ..., f(n). Suppose that an expression h satisfies $W(h) \cap \{W(q) \cup W(f(1)) \cup W(F(2)) \cup \ldots \cup W(f(n))\} = \emptyset$. Then, C is a minimal f-cover of q with respect to h,f(1),f(2),...,f(n).

Corollary: If $W(q) \cap Z(i) = \emptyset$, then any member of F(i) is irrelevant with respect to q.

Thus, when constructing f-covers we need only consider the partitions F(i) for which $Z(i) \cap W(q) \neq \emptyset$. Similarly, only the collections Z(i) for which $Z(i) \cap W(q) \neq \emptyset$ need be considered when constructing the disjunctive normal form of q during that construction.

We have now developed sufficient theory to examine the significance of our results with regard to file systems.

5) STDS: A GENERAL FILE SYSTEM:

Recalling our remarks in the introductory section, we are interested in the problem of information retrieval using Boolean expressions. Towards this end, we formulate a general model of a file system, called Set-theoretical Descriptive Storage (STDS).

Definition [9]: An STDS is a four-tuple $\langle I,A,D,\not{S}\rangle$ where

(i) I is a finite collection of undefined items called records.

(ii) A is a collection of attributes such that each a in A has a value of true (1) or false (0), denoted $E[r,a]$ for a record r in I.

(iii) D is finite collection of expressions the variables in which are members of A.

(iv) $\not{S}$ is a finite subset of the power set of I defined as follows. For any expression d in D let $S(d) = \{r|(r\in I)(E[r,d]=1)\}$. Then, $S=\{S(d)|d\in D\}$.

Informally, the set I corresponds to a file of records. The set A is a collection of conditions (such as "employee is a secretary" in a personnel file), each of which can be determined to be either true or false with respect to any record in the file. The file is 'described' using the collections D and $\not{S}$. Each of the sets S in $\not{S}$ corresponds to a definition d in D and consists of exactly those records for which the definition is true.

As can be easily shown, the definition of an STDS includes several well-known file systems; i.e., inverted files, multilist files, the generalized file structures of Hsiao and Harary (HSIAO70), Manola and Hsiao (MANOLA73), and the canonical structure of Wong and Chiang (WONG71).

The theory of f-covers provides a powerful tool for information retrieval in the STDS environment. We may construct a minimal f-cover of a query q with respect to the collection of Boolean expressions in D. The set-theoretical operations of union ($\cup$), intersection ($\cap$), and complementation ($\bar{\ }$) may then be used to combine the sets in $\$$ to obtain the smallest superset of the required set of records. The following lemmas are used to justify this process.

lemma [7]: In the STDS $<I,A,D,\$>$, for S_1,S_2 in $\$$,

(i) $S_1 \cup S_2 = \{r|(r\in I)(E[r,D(S_1) \vee D(S_2)]=1)\}$

(ii) $S_1 \cap S_2 = \{r|(r\in I)(E[r,D(S_1) \wedge D(S_2)]=1)\}$

(iii) $\bar{S}_1 = \{r|(r\in I)(E[r,\overline{D(S_1)}]=1])\}$

lemma [8]: In the STDS $<I,A,D,\$>$, for S_1, S_2 in $\$$, if $D(S_1) \Rightarrow D(S_2)$, then $S_1 \subseteq S_2$.

Corollary: The number of records in the set corresponding to the minimal f-cover of q, with respect to $\$$, is minimum compared to all f-covers of q.

By lemma [7], the set of records corresponding to an arbitrary f-cover can be determined by performing the set-theoretical operations indicated by the expression obtained from the f-cover by substituting (simultaneously)

(i) '$\cup$' for '$\vee$'

(ii) '$\cap$' for '$\wedge$'

(iii) $\bar{S}$ in $\not{S}$ for $\overline{D(S)}$

By the corollary to lemma [8], the number of records in the set corresponding to the minimal f-cover is minimum, compared to the set for any other f-cover.

The theory about irrelevant expressions was developed in order to make the construction of f-covers an efficient process in STDS. When completely general expressions are allowed as the definitions of sets, the following algorithm is suggested for retrieval.

(i) Determine the attributes to be used in the construction of the f-cover.

(ii) Construct the disjunctive normal form of the query using the attributes in (i).

(iii) Construct the minimal f-cover for the query with respect to the set definitions in the appropriate partitions.

(iv) Simplify the resulting expression for the f-cover.

(v) Perform the indicated set operations to obtain the superset of the required set.

(vi) Select (eliminate) those records in the superset which are true (false) for the query.

Thus, we can construct the minimal f-cover, using only the necessary set definitions.

Despite the advantage of the minimal f-cover with regard to the number of records in the set which corresponds to it, the minimal f-cover may not be the best choice for a given query. For example, if the sets' contents were stored as collections of hardware addresses, at unpredictable locations, then complementation of a set may be a prohibitively expensive operation. As a result of this and other considerations we present in the following section an analysis showing how to construct f-covers constrained to several specific formats.

Simplification of f-covers may be accomplished using standard techniques for the simplification of Boolean expressions, such as the Quine-McClusky method (WOOD68). These methods, however, do not take into consideration that the expressions used as variables in the f-covers are composed of a (possibly shared) set of variables. The following lemma indicates the reductions possible using this extra information.

Lemma [9]: If $f \Rightarrow g$, then $f \vee g = g$ and $f \wedge g = f$.

6) Constrained f-covers:

We shall consider f-covers whose format will be constrained in the following manners:

(i) No complementation of the f(i) ($1 \le i < n$) is allowed in the f-cover.

(ii) No 'v' operations are allowed.

(iii) No '∧' operations are allowed.

(iv) The expression is to have property (i) and either property (ii) or (iii).

To construct an f-cover with property (iii) we proceed as follows:

(i) Construct a minimal f-cover C(q) to the expression q.

(ii) Write C(q) as an 'AND'-of-'OR' terms expression using the distribution laws for Boolean expressions.

(iii) Select one of the 'OR' terms as the required f-cover.

By lemma [3] and the definition of an f-cover, the selected expression is an f-cover for q.

To construct an f-cover with property (ii), the following method can be used.

(i) Proceeding as outlined in the construction of a minimal f-cover, form the expressions h(i,j) ($1 \le i \le l, 1 \le j \le n$).

(ii) Then, the required f-cover is the 'AND' of all $h(1,j)$ $(1 \le j \le n)$ for which $h(1,j) = h(2,j) = \ldots = h(l,j)$.

If there exist $h(1,j)$ $(1 \le j \le n)$ with the property in (ii) above, then, by lemmas [2,3] and the definition of an f-cover, the expression determined is an f-cover in the required format. Similarly, by using lemma [3], we can easily demonstrate that if a non-trivial (i.e., other than the constant 1) f-cover exists, then there must exist at least one $h(1,j)$ with the property in (ii).

To construct an f-cover with property (i), we proceed as follows:

(i) As outlined in the method to construct minimal f-covers, form the expressions $h(i,j)$ $(1 \le i \le l,\ 1 \le j \le n)$.

(ii) Then, the required f-cover is given by $\bigvee_{i=1}^{\ell}$ $g(i,1)g(1,2)\ldots g(i,n)$ where $(1 \le i \le l, 1 \le j \le n)$ $g(i,j) = f(j)$ if $h(i,j) = f(j)$ and $g(i,j) = 1$ otherwise.

The method outlined above clearly produces a non-trivial f-cover in the required format if for all i $(1 \le i \le l)$ there exists at least one $j(i)$ such that $h[i,j(i)] = f(j(i))$. In the case where the minimal f-cover contains a term $\bar{f}(1)\bar{f}(2)\ldots\bar{f}(n)$, the only f-cover which does not involve complementation is the trivial f-cover consisting of the constant 1. This statement is justified by the following lemma and its corollary.

<u>Lemma[10]</u>: Let q be a Boolean expression which does not

involve complementation. Then, the disjunctive normal form of q, with respect to the variables u(1), u(2), ..., u(k), contains a term $\bar{u}(1)\bar{u}(2)\ldots\bar{u}(k)$ if and only if q is the constant 1.

<u>Corollary</u>: If the disjunctive normal form of minimal f-cover of an expression q, with respect to f(1), f(2), ..., f(n) contains a term $\bar{f}(1)\bar{f}(2)\ldots\bar{f}(n)$, then the only f-cover of q with respect to f(1), f(2), ..., f(n), which does not involve complementation, is the trivial f-cover consisting of the constant 1.

Thus, if a non-trivial f-cover which does not involve complementation of the f(j) $(1\leq j\leq n)$ exists, the method outlined above will determine one.

To derive an f-cover which involves only 'OR' operations and which does not involve complementation, we proceed as follows:

(i) Construct an f-cover with does not involve complementation.

(ii) If that f-cover is non-trivial, then write the f-cover an 'AND'-of'OR' terms expression.

(iii) Select any 'OR' term as the required f-cover.

Since the distribution laws of Boolean expressions do not involve complementation, then, using lemma [3] it is easily shown that the method determines a non-trivial f-cover in the required format if one exists.

To construct an f-cover which is the 'AND' of the uncomplemented given expressions, we proceed as follows:

(i) As outlined in the method for the construction of minimal f-covers, form the expressions $h(i,j)$ $(1 \le i \le l,\ 1 \le j \le n)$.

(ii) Then, the required f-cover is the 'AND' of all $h(i,j)$ such that $h(1,j) = h(2,j) = \ldots = h(l,j) = f(j)$.

If such a non-trivial f-cover exists, then the method determines a non-trivial f-cover in the required format, as could be shown using lemma [2].

7. Conclusions:

For a file system in which sets are used to classify the information and for which Boolean queries are used to specify retrieval requests, we have indicated:

(i) the collection of sets to be used to derive the collection of records for which the query is true;

(ii) a number of ways to combine the sets selected in (i).

Several extensions to this work are immediately apparent. For the case where a number of f-cover exist, then probability theory may be applied to select the one expected to be least costly. In the situation where experience indicates that f-covers of a certain type are preferable, then some heuristic approach may prove useful. Lastly, the general retrieval algorithm has a number of implications regarding the organization of the sets. For example, the sets' definitions and contents should be stored separately on direct-access storage media since they are never used concurrently.

APPENDIX A: Proofs of lemmas.

Lemma [4]: $C(q)$ is an f-cover of q with respect to $f(1)$, $f(2)$, $\ldots$, $f(n)$.

proof: It is sufficient to shown $q \Rightarrow C(q)$. For $1 \le i \le l$, the term $u(i,1)u(i,2)\ldots u(i,k)$ is true only for the valuation $V(i)$. By definition of $h(i,j)$ $(1 \le j \le n)$, $h(i,j)$ is true for the valuation $V(i)$. Hence, we derive $u(i,1)u(i,2)\ldots u(i,k) \Rightarrow h(i,j)$. By n applications of lemma [1] we can show $u(i,1)u(i,2)\ldots u(i,k) \Rightarrow h(i,1)h(i,2)\ldots h(i,n)$ and by l applications of lemma [1] we can show $q \Rightarrow C(q)$.

Lemma [5]: $C(q)$ is a minimal f-cover of q.

proof:

It is sufficient to show $C(q) \Rightarrow D$ for any f-cover D of q with respect to $f(1)$, $f(2)$, $\ldots$, $f(n)$. Let D be written in disjunctive normal form $\bigvee_{t=1}^{m} d(t,1)d(t,2)\ldots d(t,n)$ with respect to the n given expressions, i.e., $d(t,j)$ is $f(j)$ or $\bar{f}(j)$ for $1 \le t \le m$, $1 \le j \le m$. Now, for the i-th valuation $(1 \le i \le l)$ $V(i)$, $h(i,1)h(i,2)\ldots h(i,n)$ is true. If the t-th term of D is not identical to $h(i,1)h(i,2)\ldots h(i,n)$, then there exists a j $(1 \le j \le n)$ such that $h(i,j) = \bar{d}(t,j)$. Hence, the t-th term of D is false for the valuation $V(i)$. But, q is true for $V(i)$ and so D is true for $V(i)$. Hence, at least one term $d(t,i)d(t,2)\ldots d(t,n)$ is true for $V(i)$. Therefore, there must exist a t such that the t-th term of D is identical to the i-th term of $C(q)$, i.e., we can write $D=C(q) \vee C'$ and so $C(q) \Rightarrow D$.

<u>lemma [6]</u>: Let C be a minimal f-cover for an expression q, with respect to $f(1),f(2),\ldots f(n)$. Then, C is a minimal f-cover with respect to g, $f(1),f(2),\ldots,f(n)$ if $W[g] \cap \{W[q] \cup W[f(1)] \cup W[f(2)] \cup \ldots \cup W[f(n)]\} = \emptyset$.

<u>proof</u>:

As the proof is lengthy, we shall outline the proof and leave the details unspecified:

(1) Let P be any f-cover of q, with respect to g, $f(1)$, $f(2)$, ..., $f(n)$. Write P in disjunctive normal form as $P = \bigvee_{t=1}^{m} g(t)p(t,1)p(t,2)\ldots p(t,n)$ where $g(t)$ is g or $\bar{g}$ and where $p(t,j)$ is $f(j)$ or $\bar{f}(j)$.

(2) Show that C can be written in the form

$$C = \{\bigvee_{i=1}^{\ell} g h(i,1)h(i,2) \ldots h(i,n)\} \vee$$
$$\{\bigvee_{i=1}^{\ell} \bar{g} h(i,1)h(i,2) \ldots h(i,n)\}.$$

(3) Let V' be any valuation of the variables $W[g]$ for which g is true. Show by similar arguments to lemma [5] that there exists for all i $(1 \leq i \leq n)$ a t such that the term $gh(i,1)h(i,2)\ldots h(i,n)$ is identical to

$g(t)p(t,1)p(t,2)\ldots p(t,n)$.

(4) Similarly to (3) above, show that there exists for all i $(1 \le i \le n)$ a t such that $\bar{g}h(i,1)h(i,2)\ldots h(i,n)$ is identical to $g(t)p(t,1)p(t,2)\ldots p(t,n)$.

(5) Conclude $P = C \vee C'$ and so $C \Rightarrow P$; i.e., C is a minimal f-cover of q with respect to $g, f(1), f(2), \ldots, f(n)$.

Corollary: If $W(q) \cap Z(i) = \emptyset$, then any member of $F(i)$ is irrelevant with respect to q.

proof: The proof follows directly from lemma [5] and is omitted.

lemma [7]: In the STDS $\langle I,A,D,\$\rangle$, for S_1, $S_2 \in \$$,

(i) $S_1 \cup S_2 = \{r \mid (r \in I) \text{ and } E[r, D(S_1) \vee D(S_2)] = 1\}$

(ii) $S_1 \cap S_2 = \{r \mid (r \in I) \text{ and } E[r, D(S_1) \wedge D(S_2)] = 1\}$

(iii) $\bar{S}_1 = \{r \mid (r \in I) \text{ and } E[r, \overline{D(S_1)}] = 1\}$

proof: As the proofs for (i), (ii), and (iii) are similar we shall include only the proof for (i).

Suppose $r \in S_1 \cup S_2$. Then, $r \in S_1$ and/or $r \in S_2$. Hence, $E[r, D(S_1)] = 1$ and/or $E[r, D(S_2)] = 1$. Thus, $E[r, D(S_1) \vee D(S_2)] = 1$ and so $S_1 \cup S_2 \subseteq \{r \mid (r \in I) \text{ and } E[r, D(S_1) \vee D(S_2)] = 1\}$.

Suppose $r \in \{r | (r \in I)$ and $E[r, D(S_1) \vee D(S_2)] = 1\}$. Then, $E[r, D(S_1)] = 1$ and/or $E[r, D(S_2)\} = 1$ and so $r \in S_1 \cup S_2$, i.e., $\{r | (r \in I)$ and $E[r, D(S_1) \vee D(S_2)] = 1\} \subseteq S_1 \cup S_2$. Hence, $S_1 \cup S_2 = \{r | (r \in I)$ and $E[r, D(S_1) \vee D(S_2)] = 1\}$.

lemma [8]: In the STDS $\langle I, A, D, \not{\mathcal{S}} \rangle$, for S_1, $S_2 \in \not{\mathcal{S}}$, if $D(S_1) \Rightarrow D(S_2)$, then $S_1 \subseteq S_2$.

proof: Let $r \in S_1$. Then, $E[r, D(S_1)] = 1$. Because $D(S_1) \Rightarrow D(S_2)$, we have $E[r, D(S_2)] = 1$ and so $S_1 \subseteq S_2$.

Corollary: The number of records in the set corresponding to the minimal f-cover of q, with respect to S, is minimum compared to all f-covers of q.

proof: The proof of this corollary follows directly from lemma [8].

lemma [9]: If $f \Rightarrow g$, then $f \vee g = g$ and $f \wedge g = f$.

proof: The proof follows directly from lemmas [2,3] and is omitted.

lemma [10]: Let E be a Boolean expression which does not involve complementation. Then, the disjunctive normal form of E, with respect to the variables $u(1)$, $u(2)$, ..., $u(k)$, contains a term $\bar{u}(1)\bar{u}(2)\ldots\bar{u}(k)$ if and only if E is the constant 1.

proof: Informally, the proof is outlined as follows: Sufficiency is trivial and necessity is proven using induction on k. The basis step (k=1) is established by considering the four possible disjunctive normal forms E may have. The induction step is outlined as follows.

(i) Using Shannon's expansion theorem, factor E into two terms involving u(1) and $\bar{u}$(1).

(ii) By induction, the term involving u(1) reduces to u(1).

(iii) Hence, E reduces to the form E = u(1) u(1)E_1 where E_1 does not involve the variable u(1).

(iv) Since E can be written without complementation, E_1 must be identical to 1 since this is the only way $\bar{u}$(1) can be eliminated from the expression.

(v) Hence E = 1.

Corollary: If the disjunctive normal form of an expression q, with respect to f(1), f(2), ..., f(n), contains a term $\bar{f}$(1)$\bar{f}$(2)...$\bar{f}$(n), then the only f-cover of q, with respect to f(1), f(2), ..., f(n), which does not involve complementation, is the trivial f-cover consisting of the constant 1.

proof: Let C be a minimal f-cover of q, with respect to f(1), f(2), ..., f(n) and suppose that the disjunctive normal form of C contains a term

$\overline{f}(1)\overline{f}(2)\ldots\overline{f}(n)$. Let D be any f-cover of q, with respect to $f(1)$, $f(2)$, ..., $f(n)$, such that none of the $f(i)$ $(1 \leq i \leq n)$ is complemented.

Recalling the proof of lemma [5], we showed that D can be written as $D = C \vee C'$. Hence, the disjunctive normal form of D contains a term $\overline{f}(1)\overline{f}(2)\ldots\overline{f}(n)$ and by lemma [10] must therefore be the constant 1.

APPENDIX B: NOTATION

(1) Boolean expressions: f, g, q, h, p

(2) Boolean variables: a, b, c, d, u

(3) f-covers: C, C'

(4) sets of Boolean variables: U, W

(5) sets of valuations of Boolean variables: V

(6) equivalence relation: θ

(7) Equivalence classes: F, Z

(8) STDS: $\langle I, A, D, \not{Z} \rangle$

(9) evaluation function: E

BIBLIOGRAPHY

HSIAO70: Hsiao, D., and Harary, F. "A formal system for information from files" Communications of the ACM, Vol. 13, No. 2, (February 1970)

MANOLA73: Manola, F., and Hsiao, D "A model for keyword based file structures and access" NRL Memorandum report 2544 (distributed by NTIS: AD-745-409) (January 1973)

WOOD68: Wood, P.E., "Switching theory", McGraw-Hill Book Co., (1968)

WONG71: Wong, E., and Chiang, T.C. "Canonical file structure in attribute based file structure" Communications of the ACM, Vol. 14, No. 9, (September 1971)