THE UNIVERSITY OF MICHIGAN

Technical Report 3

CONCOMP

March 1968

DESCRIPTION OF A SET-THEORETIC DATA STRUCTURE

David L. Childs

28

THE UNIVERSITY OF MICHIGAN

Technical Report 3

DESCRIPTION OF A SET-THEORETIC DATA STRUCTURE

David L. Childs

CONCOMP: Research in Conversational Use of Computers
ORA Project 07449
F.H. Westervelt, Director

supported by:

DEPARTMENT OF DEFENSE
ADVANCED RESEARCH PROJECTS AGENCY
WASHINGTON, D.C.

CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

March 1968

THE UNIVERSITY OF MICHIGAN

Technical Report 3

DESCRIPTION OF A SET-THEORETIC DATA STRUCTURE

David L. Childs

CONCOMP: Research in Conversational Use of Computers
ORA Project 07449
F.H. Westervelt, Director

supported by:

DEPARTMENT OF DEFENSE
ADVANCED RESEARCH PROJECTS AGENCY
WASHINGTON, D.C.

CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

March 1968

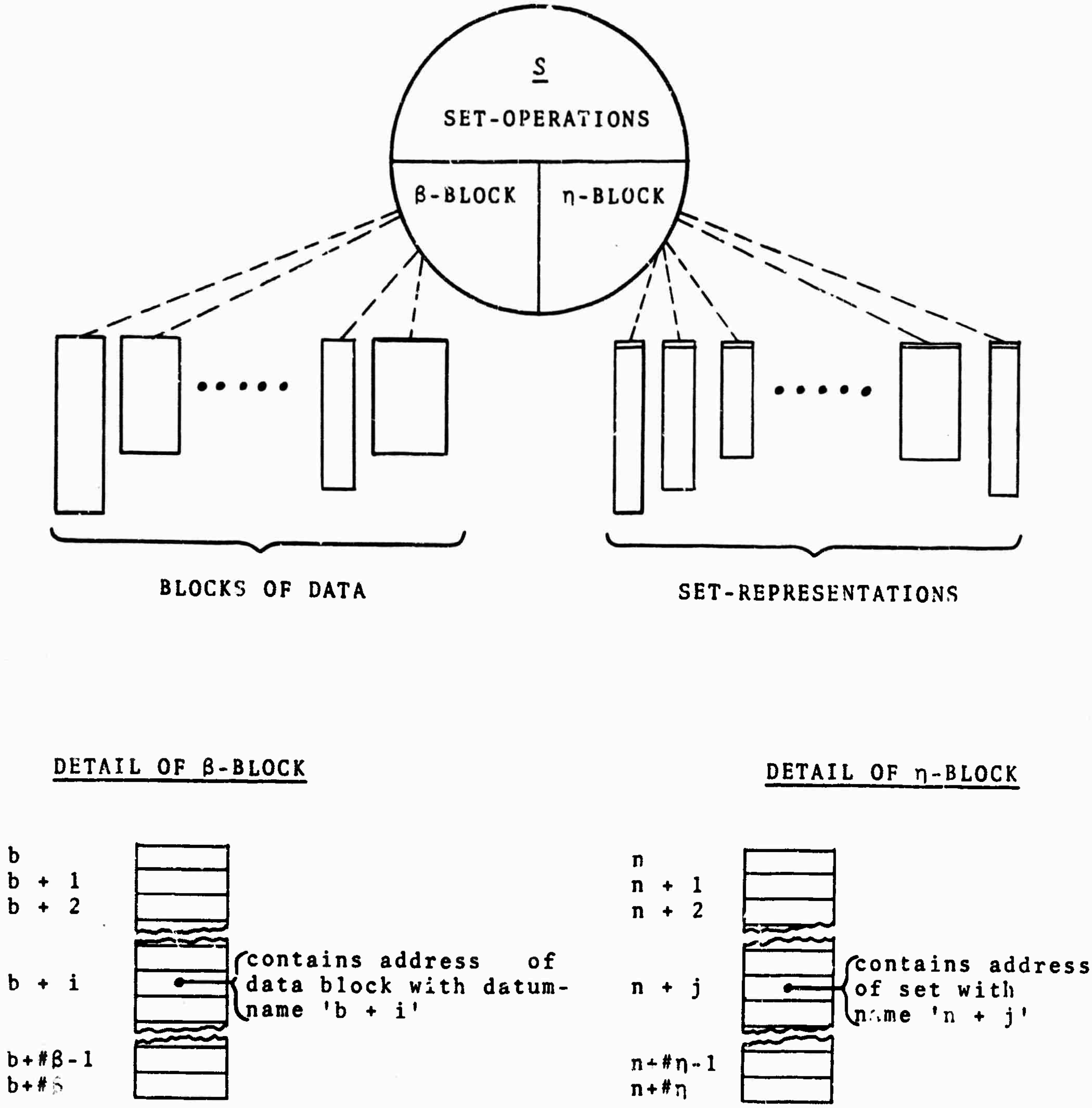
ABSTRACT

A set-theoretic data structure (STDS) is virtually a 'floating' or pointer-free structure allowing quicker access, less storage, and greater flexibility than fixed or rigid structures that rely heavily on internal pointers or hash-coding, such as 'associative or relational structures,' 'list structures,' 'ring structures,' etc. An STDS relies on set-theoretic operations to do the work usually allocated to internal pointers. A question in an STDS will be a set-theoretic expression. Each set in an STDS is completely independent of every other set, allowing modification of any set without perturbation of the rest of the structure; while fixed structures resist creation, destruction, or changes in data. An STDS is essentially a meta-structure, allowing a question to 'dictate' the structure or data-flow. A question establishes which sets are to be accessed and which operations are to be performed within and between these sets. In an STDS there are as many 'structures' as there are combinations of set-theoretic operations; and the addition, deletion, or change of data has no effect on set-theoretic operations, hence no effect on the 'dictated structures.' Thus in a floating structure like an STDS the question directs the structure, instead of being subservient to it.

BLANK PAGE

A set-theoretic data structure (STDS) is comprised of two parts: a collection of sets $\mathcal{Q}$ and a collection of set-operations S. The collection $\mathcal{Q}$ consists of two special sets, η and β, plus a finite number of other sets. The sets of $\mathcal{Q}$ are represented by blocks of contiguous storage locations with the set η containing names of all the sets, while the set β is the set of all 'datum-names.' β is represented by a contiguous block of storage locations; the address of a location in the β-block is a <u>datum-name</u> and an element of β. The content of a location in the β-block is the address of a stored description of that datum (see Fig. 1). The contents of the η-block and the β-block are the <u>only</u> pointers needed for the operation of an STDS. The storage representations of the remaining sets <u>do</u> <u>not</u> contain pointers, but contain datum-names. An STDS is a 'floating' structure or a meta-structure in the sense that the set-operations S act as the structural ties instead of using internal pointers or hash-coding. The set-operations are dependent <u>only</u> on the set η, the set containing the names of each set. Thus for any collection $\mathcal{Q}$ the set-operations are independent of: 1) the deletion or addition of datum-names, 2) any changes in datum-names, 3) the order in which the datum-names are stored, 4) the size of any set, or 5) any other modification, including the creation or deletion of sets, as long as η is kept current. Furthermore, each set in $\mathcal{Q}$ is <u>completely</u> <u>independent</u> of any other set in $\mathcal{Q}$ ($\mathcal{Q}$ need not be disjointed).

STORAGE SCHEMATIC



$\beta = \{b, b+1, b+2, \ldots, b+\#\beta\}$

$\eta = \{n, n+1, n+2, \ldots, n+\#\eta\}$

(b & n are the addresses of the initial locations of β & η respectively)

FIGURE 1

Since each set is an entity unto itself, completely free of internal pointers, and since the set-operations S are dependent only on η, the names of these sets, an STDS is relieved from the serious rigidity and excess storage encountered in fixed structures, such as 'associative or relational structures,' 'list structures,' 'ring structures,' or any other structure relying heavily on internal pointers or hash-coding.

The viability of an STDS rests on the speed and scope of the set-operations in S. The algorithms for these operations will be presented in a forthcoming paper [4]; the feasibility of the operations' being extended to sets of arbitrary length n-tuples is expressed in another paper [3] which was submitted to IFIP Congress '68. The present paper presents the available operations along with some times experienced on an IBM 7090 (see Table 1). The set-theoretic definitions appear in Appendix I, for those who are not familiar with the definitions or are not accustomed to the notation preferred in this monograph. The following tableau presents the available set operations for constructing questions in any way compatible with the parent language.

S: THE COLLECTION OF AVAILABLE SET-OPERATIONS

1) UNION

D= UN.(A,B,C)	D = {C}	$C = A \cup B$
D= UN.(1,A,C)	"	$C = \bigcup A$

2) INTERSECTION

D= IN.(A,B,C)	"	$C = A \cap B$
D= IN.(1,A,C)	"	$C = \bigcap A$

3) SYMMETRIC DIFFERENCE

D= SD.(A,B,C)	"	$C = A \blacktriangle B$
D= SD.(1,A,C)	"	$C = \Delta A$

4) RELATIVE COMPLEMENT

D= RL.(A,B,C)	"	$C = A \sim B$

5) EXACTLY N elements of A

D= EX.(N,A,C)	"	$C = E_n A$

6) DOMAIN of A

D= DM.(A,C)	"	$C = \mathcal{D}(A)$

7) RANGE of A

D= RG.(A,C)	"	$C = \mathcal{R}(A)$

8) IMAGE of B under A

D= IM.(A,B,C)	"	C = A[B]

9) CONVERSE IMAGE of B under A

D= CM.(A,B,C)	"	C = [B]A

10) CONVERSE of A

D= CV.(A,C)	"	$C = \overline{A}$

11) RESTRICTION of A to B

D= RS.(A,B,C)	"	$C = A \mid B$

12) RELATIVE PRODUCT of A and B

D= RP.(A,B,C)   D = {C}   C = A/B

13) CARTESIAN PRODUCT of A and B

D= XP.(A,B,C)   "   C = A x B

14) DOMAIN CONCURRENCE of A relative to B

D= DC.(A,B,C)   "   C = $\mathfrak{D}$(A:B)

15) RANGE CONCURRENCE of A relative to B

D= RC.(A,B,C)   "   C = $\mathfrak{R}$(A:B)

16) SET CONCURRENCE of A relative to B

D= SC.(A,B,C)   "   C = $\mathfrak{S}$(A:B)

17) CARDINALITY of A

N= C.(A)   N is a number   N = #A

<u>BOOLEAN</u> <u>OPERATIONS</u>:   Iϵ{0,1}

18) I= SBS.(A,B)   I = 1 iff   A is a subset of B

19) I= EQL.(A,B)   "   A is equal to B

20) I= DSJ.(A,B)   "   A and B are disjoint

21) I= ELM.(A,B)   "   A is an element of B

22) I= EQV.(A,B)   "   #A is equal to #B

<u>SPECIAL</u> <u>CONTROL</u> <u>OPERATIONS</u>

23) SET CONSTRUCTION

X= S.(A,B,C,D,....)   X = {A}   A = {B,C,D,...}

24) MODE of A

N= M.(A)   N ϵ{1,2,...,8}   (see text)

25) INITIAL SETTING of A

ISET.(A)   sets A to be empty or the universe depending on the function which uses it first, see Appendix II.

26) ACCESS DATA in A by FORMAT n

D= ACC.(n,A,C) n ∈{1,2,3,...} D={C}

C = may be a set of datum-names or a set of data; these two may be distinguished by the mode of C

The operations are presented in a format compatible with MAD, and with FORTRAN if the periods are removed. In general the last parameter can be deleted from any function. This default case assigns a temporary storage block, the name of which is returned by the subroutine. For example: D=UN.(A,B) gives a name in D for the temporary storage block containing the union of A and B. Since all functions operate on just the name of a storage block representing a set, and since all functions return a name, any degree of nesting of operations within operations is allowable. Two exceptions to the above are (17) and (24) which are numbers and not storage locations. In the case of (23), if only one set is given, the set is unchanged, but the name of the set is put in X. The <u>MODE</u> of a set is covered in depth in an aforementioned paper [4]. It will suffice here to explain that 'mode' represents one of eight different storage configurations, each tailored to special sets and operations. The functions <u>do not require</u> participating sets to be of the same mode. Notice that all the operations are defined for <u>any</u> set though the result in some cases may always be empty as in the case of DM.(A) where A is the set of the first 10,000 integers. A forthcoming paper [2] will show that there is a meaningful definition for relations covering arbitrary sets of variable length n-tuples without couching these relations as sets of ordered pairs. Also, the binary-relation properties (e.g., domain, image, relative product, restriction, etc.) are extended in a meaningful way to cover this extended concept of relation. These extended operations can also be implemented in an STDS [3].

Table 1 gives the results of implementing some of these operations on the IBM 7090. The four operations considered here are: unary union, unary intersection, unary symmetric difference, and 'exactly n' for n ε{1,...,#G} where G is the family of sets being operated on. The number of elements in G is given by #G. All the elements of G contain the same number of elements, #A, and the size of the population which the elements of each A were chosen from is #P.

It should be noted that the times in Table 1 are dependent on the total number of elements contained in the elements in G, and <u>not</u> the number of elements in G. In (d) through (i) the total number of elements contained in the elements of G is 10,000. While #G varies from 20 to 500, the times for UN. and SD. remain the same.

	#P	#G	#A	OPERATION	SECONDS
a)	500	2	200	UN.(1,G)	.03
				IN.(1,G)	.05
				SD.(1,G)	.03
				EX.(1,G) to EX.(2,G)	.16
b)	500	4	200	UN.(1,G)	.06
				IN.(1,G)	.12
				SD.(1,G)	.06
				EX.(1,G) to EX.(4,G)	.42
c)	50	10	20	UN.(1,G)	.01
				IN.(1,G)	.10
				SD.(1,G)	.03
				EX.(1,G) to EX.(10,G)	.37
d)	1200	10	1000	UN.(1,G)	.73
				IN.(1,G)	.90
				SD.(1,G)	.76
				EX.(1,G) to EX.(10,G)	7.89
e)	1000	20	500	UN.(1,G)	.73
				IN.(1,G)	.48
				SD.(1,G)	.76
				EX.(1,G) to EX.(18,G)	10.96
f)	1000	50	200	UN.(1,G)	.75
				IN.(1,G)	.16
				SD.(1,G)	.76
				EX.(1,G) to EX.(20,G)	11.00
g)	1000	100	100	UN.(1,G)	.75
				IN.(1,G)	.15
				SD.(1,G)	.76
				EX.(1,G) to EX.(23,G)	11.88
h)	1000	200	50	UN.(1,G)	.75
				IN.(1,G)	.06
				SD.(1,G)	.78
				EX.(1,G) to EX.(24,G)	12.36
i)	1000	500	20	UN.(1,G)	.76
				IN.(1,G)	.05
				SD.(1,G)	.78
				EX.(1,G)	12.50

Table 1. EXECUTION TIMES FOR SET OPERATIONS ON THE IBM 7090

The rest of this paper will be devoted to examples demonstrating the applicability of an STDS.

EXAMPLE 1

Let there be six sets: A,B,C,D,E,F,the membership lists of six country clubs. For each male resident of Ann Arbor, let there be a datum-name in β for a data-block containing: person's name, address, phone number, credit rating, age, golf handicap, wife's name (if any), political affiliation, religious preference, and salary. The set η will contain the names of the sets, namely: A(0), B(0), C(0), D(0), E(0), F(0). This along with the collection S of set operations allows answering the following questions.

1) How many members belong to club A or B but not C?
2) Find the phone numbers of members in an odd number of clubs.
3) Get addresses of members belonging to one and only one club.
4) Get addresses and phone numbers of people not in any club.
5) Find members of A that are not also in B but who may be in C only if they are not in D, or in E if they are not in F.
6) Get the average credit rating of members belonging to exactly three clubs.

The possible questions may become ridiculously involved and may interact with any spontaneously constructed sets. For example of the latter, let X be the set of Ann Arbor males born in Ann Arbor.

7) Find the average age of members born in Ann Arbor and compare with average age of members not born in Ann Arbor.

The answers to (1) through (7) formulated in an STDS are expressed below, with N and M representing real numbers, and with BB for β and NN for η.

1) N = C.(RL.(UN.(A,B),C))

ans: N

2) ACC.(1,SD.(1,NN),Q)

ans: Q   Format 1 gives phone numbers

3) ACC.(2,EX.(1,NN),Q)

ans: Q   Format 2 gives addresses

4) ACC.(3,RL.(BB,UN.(1,NN)),Q)

ans: Q   Format 3 gives phone numbers and addresses

5) RL.(RL.(A,B),UN.(RL.(D,C),RL.(F,E)),Q)

ans: Q

6)
```
     ACC.(4,EX.(3,NN),Q)
     N = 0
     THROUGH LOOP, FOR I = 1,1,I.G.C.(Q)
LOOP N = N + Q(I)
     N = N/C.(Q)
```

ans: N   Format 4 gives credit rating

7)
```
            N = 0
            M = 0
            ACC.(5,X,T)
            THROUGH LOOP1,FOR I=1,1,I.G.C.(T)
   LOOP1    N = N + T(I)
            ACC.(5,RL.(BB,X),P)
            THROUGH LOOP2, FOR I=1,1,I.G.C.(P)
   LOOP2    M = M + P(I)
            N = N/C.(T)
            M = M/C.(P)
```

ans: N and M are the respective average ages

Format 5 gives ages

EXAMPLE 2

Family lineage is easily expressed in an STDS. With just five initial relations defined over a population U, all questions concerning family ties may be expressed.

Let U be a population of people and let

M = {<x,y>: y is the mother of x}

F = {<x,y>: y is the father of x}

S = {<x,y>: y is a sister of x}

B = {<x,y>: y is a brother of x}

H = {<x,y>: y is a husband of x}

Let X be any subset of the population U, find

1) the set G of Grandfathers of X.

G = F[(F ∪ M)[X]]	set notation
IM.(F,IM.(UN.(F,M),X),G)	in an STDS

2) the set GF of Grandfathers of X on the father's side.

GF = F[F[X]]	set notation
IM.(F,IM,(F,X),GF)	STDS

3) the set GM of Grandfathers of X on the mother's side.

$GM = G \sim GF$	set notation
RL.(G,GF,GM)	STDS

4) the set GR: the grandfather relations over U.

$GR = (F \cup M)/F$	set notation
RP.(UN.(F,M),F,GR)	STDS

5) the general relation: P = {<x,y>: y is a parent of x}

$P = F \cup M$	set notation
UN.(F,M,P)	STDS

6) the general relation: Sibling, L.

$L = S \cup B$	set notation
UN.(S,B,L)	STDS

7) the general relation: Children, C.

$C = \overline{M \cup F} = \overline{P}$	set notation
CV.(P,C)	STDS

8) the general relation: Aunt, A.

$A = (P/S) \cup (P/B/\overline{H})$	set notation
UN.(RP.(P,S),RP.(P,RP.(B,CV.(H))),A)	STDS

9) the general relation: Wife, W.

$W = \overline{H}$	set notation
CV.(H,W)	STDS

10) the general relation: Cousin, K.

$K = P/L/C$	set notation
RP.(P,RP.(L,C),K)	STDS

11) the general relation: Half-sibling, HS.

$HS = P/C \sim (M/\overline{M} \cap F/\overline{F})$ set notation

```
RL.(RP.(CV.(C),C),IN.(RP.(M,CV.(M)),
     RP.(F,CV.(F))),HS)                  STDS
```

12) people in X with no brothers or sisters

$Q = X \sim \mathcal{D}(L)$ set notation

```
RL.(X,DM.(L),Q)                          STDS
```

13) find all relations of X to a set Y such that Y is equal to the image of X.

$Q = \{A : (A \epsilon \eta)(Y = A[X])\}$ set notation

```
      ISET.(Q)                           STDS
      DC.(X,NN,T)
      THROUGH LOOP, FOR I=1,1,I.G.C.(T)
      B = IM.(T(I),X)
LOOP  WHENEVER EQL.(Y,B).E.1, UN.(Q,S.(T(I)),Q)
```

Many more possibilities are available and might be tried by the reader.

An example of quantified questions will be found in Appendix II, which may also be of help to the reader who is familiar with associative data structures. Also of interest is a recently completed implementation of an associative data structure [1], which, while not as general as an STDS, is more general than other known implemented data structures.

APPENDIX I

SET-THEORETIC DEFINITIONS

Conventions

The logical connectives 'and,' 'or,' 'exclusive-or' are represented by '$\wedge$,' 'v,' '$\triangle$.' 'For all x,' 'for some x,' 'for exactly n x' will be represented by '$\forall$x,' '$\exists$x,' 'E(n)!x.' Parentheses are used for separation, and as usual the concatenation of parentheses will represent conjunction.

'A' will be a <u>set</u> if and only if (a) it can be represented formally by abstraction (i.e., $A=\{x:\theta(x)\}$ where $\theta(x)$ is a predicate condition specifying the allowable elements 'x'); (b) 'A' can be represented by {,} enclosing the specific elements of 'A.'

Definitions

The symbol 'e' means 'is an element of'; xeA reads: "x is an element of A."

1) UNION

a) binary union of two sets A and B

$$A \cup B = \{x:(x \in A) \vee (x \in B)\}$$

b) unary union of a family G of sets

$$\bigcup G = \{x:(\exists A \in G)(x \in A)\}$$

c) indexed union of a set f(A) over the family G

$$\bigcup_{A \in G} f(A) = \{x:(\exists A \in G)(x \in f(A))\}$$

PRECEDING PAGE BLANK

2) INTERSECTION

a) binary intersection of A and B

$$A \cap B = \{x:(x\epsilon A)(x\epsilon B)\}$$

b) unary intersection of a family G

$$\bigcap G = \{x:(\forall A\epsilon G)(x\epsilon A)\}$$

c) indexed intersection of f(A) over the family G

$$\bigcap_{A\epsilon G} f(A) = \{x:(\forall A\epsilon G)(x\epsilon f(A))\}$$

3) SYMMETRIC DIFFERENCE

a) binary symmetric difference of A and B

$$A \blacktriangle B = \{x:(x\epsilon A)\blacktriangle(x\epsilon B)\}^*$$

*even though the symbol '▲' has two different meanings, no confusion is likely

b) unary symmetric difference of G

$$\Delta G = \{x:(\text{for an odd number of } A\epsilon G)(x\epsilon A)\}$$

c) indexed symmetric difference of f(A) over G

$$\Delta_{A\epsilon G} f(A) = \{x:(\text{for odd no. of } A\epsilon G)(x\epsilon f(A))\}$$

4) RELATIVE COMPLEMENT

$$A \sim B = \{x:(x\epsilon A)(x\not\epsilon B)\}$$

5) EXACTLY N!

the set of elements common to exactly 'n' elements of

a given set G is represented by:

$$E_n G = \{x:(E(n)!A\epsilon G)(x\epsilon A)\}$$

6) DOMAIN of a set A

$$\mathcal{D}(A) = \{x:(\exists y)(\langle x,y\rangle \epsilon A)\}^*$$

*<x,y> represents an ordered pair

7) RANGE of a set A

$$\mathcal{R}(A) = \{y:(\exists x)(\langle x,y\rangle \epsilon A)\}$$

8) IMAGE of B under A

$$A[B] = \{y : (\exists x \in B)(\langle x,y\rangle \in A)\}$$

9) CONVERSE IMAGE of B under A

$$[B]A = \{x : (\exists y \in B)(\langle x,y\rangle \in A)\}$$

10) CONVERSE of A

$$\bar{A} = \{\langle y,x\rangle : \langle x,y\rangle \in A\}$$

11) RESTRICTION of A to B

$$A|B = \{\langle x,y\rangle : (\langle x,y\rangle \in A)(x \in B)\}$$

12) RELATIVE PRODUCT of A and B

$$A/B = \{\langle x,y\rangle : (\exists z)(\langle x,z\rangle \in A)(\langle z,y\rangle \in B)\}$$

13) CARTESIAN PRODUCT of A and B

$$A \times B = \{\langle x,y\rangle : (x \in A)(y \in B)\}$$

14) DOMAIN CONCURRENCE of X relative to A

$$\mathfrak{D}(X:A) = \{B : (B \in A)(X \in \mathcal{D}(B))\}$$

15) RANGE CONCURRENCE of X relative to A

$$\mathfrak{R}(X:A) = \{B : (B \in A)(X \in R(B))\}$$

16) SET CONCURRENCE of X relative to A

$$\mathfrak{S}(X:A) = \{B : (B \in A)(X \in B)\}$$

17) CARDINALITY of A

#A = n iff there are exactly n elements in A

18) A is a SUBSET of B iff every element of A is an element of B

19) A is EQUAL to B iff A is a subset of B, and B is a subset of A

20) A and B are DISJOINT iff the intersection of A and B is empty

21) A is EQUIVALENT to B iff A and B contain the same number of elements

APPENDIX II

TRANSFORMULATION OF AN ASSOCIATIVE DATA STRUCTURE

If, in J.A. Feldman's paper [5], an 'attribute' represents a relation, then since any relation can be represented by a set of ordered-pairs, the formulation involving ordered triples may be abandoned in favor of sets of ordered-pairs. A correspondence may then be made between the expression A(o)=v and a set-theoretic interpretation. In Feldman's paper six questions are represented by: A(o)=?, A(?)=v, ?(o)=v, A(?)=?, ?(o)=?, and ?(?)=v. As presented in the paper the expressions are ambiguous concerning whether 'o' and 'v' represent sets, or elements, or both. The general formulation is to assume that they are sets, and to replace 'o' and 'v' by the sets 'X' and 'Y', and to replace A by a set of relations R. If the original intention was for 'o' and 'v' to be elements, then X and Y will just be singleton sets. '<u>R(X)=Y</u>' is now the general form, and generation of questions is accomplished by asserting one or two of the three sets and pondering the remaining. Just deleting one or two sets, however, does not yield a well-formed question; many interpretations may be possible. In an STDS all interpretations may be made explicit. For a sampling, each of the six questions is formulated in the most general way and then in some less general interpretations.

1) R(X) = Q

Given a set of relations R and a set of elements X, find Q. The most general interpretation for Q is: Find the set of elements 'v' such that <o,v>eA

a) for some AeR and some oeX

$$Q = \bigcup_{AeR} \bigcup_{oeX} A[\{o\}] = \{v:(\exists AeR)(\exists oeX)(oAv)\}$$

Less general interpretations may be given Q by replacing quantifiers or changing their order:

b) for all AeR and exactly one oeX

$$Q = \bigcap_{AeR} E(1)_{oeX} A[\{o\}] = \{v:(\forall AeR)(E(1)!oeX)(oAv)\}$$

c) for some oeX and all AeR

$$Q = \bigcup_{oeX} \bigcap_{AeR} A[\{o\}] = \{v:(\exists oeX)(\forall AeR)(oAv)\}$$

d) for all oeX and for an odd number of AeR

$$Q = \bigcap_{oeX} \Delta_{AeR} A[\{o\}] = \{v:(\forall oeX)(\Theta AeR)(oAv)\}$$

e) for some AeR and all oeX

$$Q = \bigcup_{AeR} \bigcap_{oeX} A[\{o\}] = \{v:(\exists AeR)(\forall oeX)(oAv)\}$$

Expressed in an STDS, these questions become:

a)
```
              ISET.(Q)
              THROUGH LOOP, FOR I=1,1,I.G.C.(R)
     LOOP     UN.(Q,IM.(R(I),X),Q)
```

b)
```
              ISET.(Q)
              THROUGH LOOP, FOR I=1,1,I.G.C.(R)
              ISET.(T)
              THROUGH LOOP. FOR J=1,1,J.G.C.(X)
     LOOP     IN.(Q,EX.(1,IM,(R(I),X(J)),T).Q)
```

c)
```
              ISET.(Q)
              THROUGH LOOP, FOR I=1,1,I.G.C.(X)
              ISET.(T)
              THROUGH LOOP, FOR J=1,1,J.G.C.(R)
     LOOP     UN.(Q,IN.(T,IM.(R(J),X(T)),T),Q)
```

d)

```
              ISET.(Q)
              THROUGH LOOP, FOR I=1,1,I.G.C.(X)
              ISET.(T)
              THROUGH LOOP, FOR J=1,1,J.G.C.(R)
     LOOP     IN.(Q,SD.(T,IM.(R(J),X(I)),T),Q)
```

e)

```
              ISET.(Q)
              THROUGH LOOP, FOR I=1,1,I.G.C.(R)
              ISET.(T)
              THROUGH LOOP, FOR J=1,1,J.G.C.(X)
     LOOP     UN.(Q,IN.(T,IM.(R(I),X(J)),T),Q)
```

2) R(Q) = Y

Given a set of relations R and a set of elements Y, find Q. Just the most general interpretation will be given since quantifier manipulation was demonstrated by (1). Find the set of elements 'o' such that <o,v>eA for any AeR and any veY

$$Q = \bigcup_{A\epsilon R} \bigcup_{v\epsilon Y} [\{v\}]A = \{o:(\exists A\epsilon R)(\exists v\epsilon Y)(oAv)\}$$

gives

```
              ISET.(Q)
              THROUGH LOOP, FOR I=1,1,I.G.C.(R)
              ISET.(T)
              THROUGH LOOP, FOR J=1,1,J.G.C.(Y)
     LOOP     UN.(Q,UN.(T,CM.(R(I),Y(J)),T),Q)
```

3) Q(X) = Y

Given two sets X and Y find the set of relations A such that <o,v>eA for some oeX and some veY

$$Q = \bigcup_{o\epsilon X} \bigcup_{v\epsilon Y} \mathfrak{S}(\{\langle o,v\rangle\}) = \{A:(\exists o\epsilon X)(\exists v\epsilon Y)(oAv)\}$$

gives

```
              ISET.(Q)
              THROUGH LOOP, FOR I=1,1,I.G.C.(X)
              ISET.(T)
              THROUGH LOOP, FOR J=1,1,J.G.C.(Y)
     LOOP     UN.(Q,UN.(T,SC.(XP.(X(I),Y(J))),T),Q)
```

4) $R(Q_o) = Q_v$

Given a set of relations R there is no obvious delineation of sets Q_o or Q_v; three generically different questions may be phrased, each one of which may be expressed in different degrees of generality.

i) Find Q_o independent of Q_v

ii) Find Q_v independent of Q_o

iii) Find $Q_o \times Q_v$

For (i) find the set of 'o' such that for some A in R there exists a 'v' such that <o,v>eA

$$Q_o = \bigcup_{A\epsilon R} \mathcal{D}(A) = \{o:(\exists A\epsilon R)(o\epsilon\mathcal{D}(A))\} = \{o:(\exists A\epsilon R)(\exists v\epsilon\beta)(oAv)\}$$

For (ii) find the set of 'v' such that for some A in R there exists an 'o' such that <o,v>eA

$$Q_v = \bigcup_{A\epsilon R} \mathcal{R}(A) = \{v:(\exists A\epsilon R)(v\epsilon\mathcal{R}(A))\} = \{v:(\exists A\epsilon R)(\exists o\epsilon\beta)(oAv)\}$$

For (iii) find the set of <o,v> such that for some A in R <o,v>eA

$$Q = \bigcup_{A\epsilon R} A = \{\langle o,v\rangle:(\exists A\epsilon R)(oAv)\}$$

These are represented in an STDS by:

```
   i)           ISET.(Q)
                THROUGH LOOP, FOR I=1,1,I.G.C.(R)
        LOOP    UN.(Q,DM.(R(I)),Q)

  ii)           ISET.(Q)
                THROUGH LOOP, FOR I=1,1,I.G.C.(R)
        LOOP    UN.(Q,RG.(R(I)),Q)

 iii)           Q = UN.(1,R,Q)
```

5) $Q_A(X) = Q_v$

Given a single set X requires, as in (4), three separated formulations:

i) Find Q_A independent of Q_v

ii) Find Q_v independent of Q_A

iii) Find $Q_A \times Q_v$

For (i) find set of 'A' such that for some oeX there exists a 'v' such that <o,v>eA

$$Q_A = \bigcup_{o \in X} \mathfrak{D}(\{o\}:\eta) = \{A:(\exists o \in X)(o \in \mathcal{D}(A))\}$$
$$= \{A:(\exists o \in X)(\exists v \in \beta)(oAv)\}$$

For (ii) find set of 'v' such that for some oeX there exists an 'A' such that <o,v>eA

$$Q_v = \bigcup_{o \in X} \bigcup_{A \in \eta} A[\{o\}] = \{v:(\exists o \in X)(\exists A \in \eta)(oAv)\}$$
$$= \{v:(\exists o \in X)(\exists A \in \mathfrak{D}(X:\eta))(v \in A[\{o\}])\}$$

For (iii) find the set of <A,v> such that for some oeX <o,v>eA

$$Q = \bigcup_{o \in X} \bigcup_{A \in \eta} \{A\} \times A[\{o\}] = \{\langle A,v\rangle:(\exists o \in X)(oAv)\}$$
$$= \{\langle A,v\rangle:(\exists o \in X)(A \in \mathfrak{D}(X:\eta))(v \in A[\{o\}])\}$$

These are expressed in an STDS as:

```
     i)           DC.(X,NN,Q)

    ii)           ISET.(Q)
                  THROUGH LOOP, FOR I=1,1,I.G.C.(X)
                  DC.(X(I),NN,A)                       (see note)
                  THROUGH LOOP, FOR J=1,1,J.G.C.(A)
         LOOP     UN.(Q,IM.(A(J),X(I)),Q)

   iii)           ISET.(Q)
                  THROUGH LOOP, FOR I=1,1,I.G.C.(X)
                  DC.(X(I),NN,A)                       (see note)
                  THROUGH LOOP, FOR J=1,1,J.G.C.(A)
         LOOP     UN.(Q,XP.(S.(A(J)),IM.(A(J),X(I))),Q)
```

NOTE: Execution is minimized since $C.(A) \neq C.(NN)$ and the substitution of $\bigcup_{o \in X} \bigcup_{A \in \mathfrak{D}(\{o\}:\eta)}$ for $\bigcup_{o \in X}$ '' $_{A \in \eta}$ is justified by a trivial theorem [3] which states: given X and η then

$$(\forall o \in X)(\forall A \in \eta)(A \in \mathfrak{D}(\{o\}:\eta) \leftrightarrow A[\{o\}] \neq \emptyset)$$

6) $Q_A(Q_c) = Y$ is similar to (5).

GLOSSARY OF SYMBOLS

Symbol	Symbol Definition
iff	if and only if
=	Identity
$\wedge$	Conjunction
$\vee$	Disjunction
$\triangle$	Exclusive or
$\rightarrow$	Implication (if ... then)
$\leftrightarrow$	Equivalence
$\forall x$	Universal quantifier (for all)
$\exists x$	Existential quantifier (for some)
E!x	Uniqueness quantifier (for exactly one)
Θx	Odd quantifier (for an odd number of)
E(n)!x	Exact number quantifier
$\in$	Set membership
$\emptyset$	Empty set
$\notin$	Non-membership
$\subset$	Set inclusion
$A \cap B$	Intersection
$A \cup B$	Union
$A \triangle B$	Symmetric difference
$A \sim B$	Relative complement
<x,y>	Ordered pair
$\{x:\theta(x)\}$	Definition by abstraction
xAy	Ordered pair <x,y> contained in A

Symbol	Symbol Definition
$\bigcup G$	Union or sum of G
$\bigcap G$	Intersection of G
ΔG	Symmetric difference of G
$E_n G$	Elements contained in exactly n elements of G
$A \times B$	Cartesian product
$\mathcal{D}(A)$	Domain of A
$\mathcal{R}(A)$	Range of A
$\bar{A}$	Converse of A
A/B	Relative product of A and B
$A\|X$	A restricted to X
$A[X]$	Image of X under A
$[X]A$	Converse-image of X under A
$\mathfrak{D}(X:A)$	Domain-concurrence of X relative to A
$\mathfrak{R}(X:A)$	Range-concurrence of X relative to A
$\mathfrak{S}(X:A)$	Set-concurrence of X relative to A

REFERENCES

1. Ash, W.L. and E.H. Sibley, TRAMP: A Relational Memory with an Associative Base, Concomp Technical Report (in preparation).

2. Childs, D.L., Reconstituted Set-Theoretic Definition for Relations, Concomp Technical Report (in preparation).

3. Childs, D.L., Feasibility of a Set-Theoretic Data Structure Based on a Reconstituted Set-Theoretic Definition for Relations, Concomp Technical Report (in preparation).

4. Childs, D.L., Implementing Set-Theoretic Operations for a Set-Theoretic Data Structure, Concomp Technical Report (in preparation).

5. Feldman, J.A., Aspects of Associative Processing, Technical Note 1965-13, M.I.T. Lincoln Laboratories, April 1965.

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) The University of Michigan CONCOMP Project	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

DESCRIPTION OF A SET-THEORETIC DATA STRUCTURE

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5. AUTHOR(S) (Last name, first name, initial)

CHILDS, David L.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1968	27	5

8a. CONTRACT OR GRANT NO. DA-49-083 OSA-3050 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report 3 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. AVAILABILITY/LIMITATION NOTICES

Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

A set-theoretic data structure (STDS) is virtually a 'floating' or pointer-free structure allowing quicker access, less storage, and greater flexibility than fixed or rigid structures that rely heavily on internal pointers or hash-coding, such as 'associative or relational structures,' 'list structures,' 'ring structures,' etc. An STDS relies on set-theoretic operations to do the work usually allocated to internal pointers. A question in an STDS will be a set-theoretic expression. Each set in an STDS is completely independent of every other set, allowing modification of any set without perturbation of the rest of the structure; while fixed structures resist creation, destruction, or changes in data. An STDS is essentially a meta-structure, allowing a question to 'dictate' the structure or data-flow. A question establishes which sets are to be accessed and which operations are to be performed within and between these sets. In an STDS there are as many 'structures' as there are combinations of set-theoretic operations; and the addition, deletion, or change of data has no effect on set-theoretic operations, hence no effect on the 'dictated structures.' Thus in a floating structure like an STDS the question directs the structure, instead of being subservient to it.

DD FORM 1 JAN 64 1473

Security Classification

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Associative data structure						
Data modification						
Datum-names						
Floating structure						
Information retrieval						
Meta-structure						
Pointer-free structure						
Quantified questions						
Set						
Set operations						
Set theoretic data structure						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ______________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ______________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ______________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (*TS*), (*S*), (*C*), or (*U*).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.