AN EFFICIENT ALGORITHM FOR FINDING THE UNION, INTERSECTION AND DIFFERENCES OF SPATIAL DOMAINS

by

C. M. Eastman and C. I. Yessios

September 1972

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213

This work was supported by the National Science Foundation under grant GJ 31188 to Carnegie-Mellon University and also by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-70-C-0107), monitored by the Air Force Office of Scientific Research.

1. Introduction

Information is often aggregated over spatial domains. The result is a set of domains with specified properties. Examples include architectural and engineering design, where a domain may be a room or physical entity [2], urban planning, where socio-economic data is aggregated by census tracts [4], and cartography [1]. Most often, a spatial domain depicts a (possibly large) point set.

Many different data structures have been used to depict spatial domains within a computer [3], [5], [6]. Perhaps one of the simplest is the closed curve. Indeed, for graphical output of domains on a plotter or cathode ray tube, the closed curve is "natural."

Fundamental operations on spatial domains are the set theoretic ones, e.g., the union, intersection, and differences. To date, no efficient algorithms have been devised for executing all these operations on spatial domains when they are represented as closed curves. Some of the operations have been implemented for other data structures, but they are computationally quite expensive [1], [6].

In this paper, we describe a single algorithm capable of deriving the union, intersection, and differences of two spatial domains, when they are represented as closed curves. The algorithm is general and can be applied to any two sets of closed curves depicting spatial domains. Multiple operations are easily executed. The presentation is organized as follows. We first introduce the necessary definitions and the properties of closed curves. The algorithm is first presented informally in sections 4, 5, and 6. It is then formally defined in section 7. Sections 8 and 9 generalize the algorithm.

2. Initial Definitions

A two dimensional spatial domain can be represented by the closed curve which delineates it. Applying discrete arithmetic, such a curve can be approximated by a point vector (an ordered set of points), which will also be called a curve and will be represented as $C^k = (P_0^k, P_1^k, \ldots, P_i^k, \ldots, P_n^k, P_0^k)$. Above, k is the index or the identifying label of the curve, i the index of the points in the curve and each $P_i^k = (x_i, y_i)$, where x and y are standard Cartesian coordinates.

P_0^k, the point listed twice, is called the <u>closing point</u> of the curve C^k. A closed curve partitions a plane into two disjoint domains, one being finite and the other infinite. Either one can be designated as the curve's <u>inner</u> space and the other as its <u>outer</u> space. A curve'e inner space is precisely the domain the curve delineates and represents. A point vector can be traced in two directions; the direction where the inner space is to the right of the line shall be referred to as the <u>positive order</u>, the direction where the inner space is to the left shall be called the <u>negative order</u>. (See Figure 2.1.)

Figure 2.1 about here

Figure 2.2 about here

A single curve in the positive direction delineates a single <u>continuous</u> domain. A domain may also be discontinuous or have a ring-like structure or some combination of these two (Figure 2.2). Domains characterized by multiple curves shall be referred to as <u>discontinuous</u>. Discontinuous curves shall constitute an ordered set of curves, referred to as the domain's <u>curve set</u>.

in general, we denote a domain as

$$D^h = \{C^{h1}, C^{h2}, \ldots, C^{hk}, \ldots, C^{hn}\}$$

where k is the index of the curves in the domain and h is the index or identifying label of the domain. When $n=1$, the above equivalence becomes $D^h = \{C^h\}$ and the domain is continuous. For continuous domains we may write simply C^h. If $n>1$, the domain is discontinuous. For most of this paper we shall be concerned with continuous domains. The discontinuous case will be taken up in section 9.

The points lying on a curve (and therefore belonging to neither the inner or outer space) will be called <u>boundary points</u>. These are of two kinds; <u>corner points</u> (to be abbreviated as p-points) and <u>flat points</u> (f-points). The p-points delineate the curve's segments and the f-points lie on the segments between and excluding the p-points. Initially, a curve is given as an ordered set of p-points, which is referred to as its point vector. Semantically, a curve does not change if we write one or more of its f-points as corner points and properly include them in its point vector. By properly we mean that each such point should be inserted between the p-points which delineate the line segment upon which it lies. The above constitutes an operation called an <u>expansion</u>.

Two (or more) domains may be <u>disjoint</u>, <u>conjoint</u>, <u>subjoint</u> or <u>coincident</u>. They are disjoint if they have no common points; conjoint if some (and only some) of their points are common; subjoint if all the points of the one domain are also points of the other domain; coincident if the two domains have exactly the same points. When two disjoint domains have common boundary points, they are more specifically called distangent; similarly subjoint domains are called subtangent if they have common boundary points. Examples are given in Figure 2.3.

<u>Figure 2.3 about here</u>

In this paper, we shall be interested in three operations; the union ($\cup$), the intersection ($\cap$) and the difference ($-$). The union and the intersection are symmetric, while the difference is not. That is if C^1 and C^2 two closed curves, then $C^1 \cup C^2 = C^2 \cup C^1$ and $C^1 \cap C^2 = C^2 \cap C^1$ while $C^1 - C^2 \neq C^2 - C^1$. Therefore, given any pair of domains, a total of four different operations are applicable. In Figure 2.4 we summarize their results. The notation $\overline{C}^k$ denotes that C^k is traced in the negative order.

Figure 2.4 about here

As can be seen, when the two curves (C^1 and C^2) are disjoint, subjoint or coincident, the derivation of their union intersection and differences is straightforward. The resulting domain is empty or is given by either one or both of the curves delineating the domains under consideration. traced in positive or negative order. The problem is more complicated when the two domains are conjoint. The resulting curves are given by a proper mixing of the points in the initial curves (See Figure 2.5).

Figure 2.5 about here

This paper focusses primarily on the latter case. It develops an algorithm which, by properly mixing the boundary points of two conjoint domains, derives the curve set which delineates the union, intersection or difference. Section 8 generalizes the algorithm to include the cases of disjoint, subjoint and coincident domains.

3. Conjoint Domains

The main characteristic of the conjoint domains, as opposed to the other cases, is that the curves intersect. The points of intersection are common to both curves. Intersections are not the only case of common points; the curves may also have tangent points which are also common for both curves. In general, any two line segments may be related in any one of four ways (shown in Figure 3.1). In case (1) the two segments have no common point; in (2) they intersect at a point which is an f-point for both segments; in (3) the segments touch at a common point which is an f-point for the one and a c-point for the other curve; in (4) the segments touch at a point which is a c-point for both curves. Case (1) is referred as a non-intersecting pair, case (2) as an intersecting pair and cases (3 and (4) as tangent pairs. In (3) we have a single sided tangency while in (4) the tangency is double sided.

<u>Figure 3.1 about here</u>

Tangent points may or may not be treated as an intersection. For this to be decided, we need to have information about all the segments to which the tangent point belongs. In case (3), three line segments are involved, while in case (4), four segments. We look into these matters in Section 4.

We call a pair of curves and their conjunction <u>regular</u>, if all their common points are f-points for both curves. This is the case when all the intersection points are f-points for both curves and the curves have no tangent points. If at least one of their common points is a corner point for at least one of the curves, we call the curves and their conjunction <u>irregular</u>.

The algorithm to be developed is based on the following properties of the conjoint regular curves:

I. The intersection points, to be denoted by S_j, are common in both curves. In addition, they are all part of the curve sets which delineate the union, the intersection and the differences.

II. The non-intersecting corner points (P_i^k) of two conjoint curves are distinguished into <u>inner</u>, to be denoted $\overline{P}_i^k$ and <u>outer</u>, to be denoted $\underline{P}_i^k$. P_i^k of a curve C^k is outer with respect to another curve $C^{k'}$ if it lies in the outer space delineated by $C^{k'}$. It is inner otherwise.

At each S point the border lines of two domains intersect. Then for each such border line an S point implies a crossing over the other domain's boundary line. Thus, if a point just before S is in the domain, a point right after it will be out of the domain. If we have two S points in a row, say S_1 and S_2, then the second negates the effect of the first. That is if a point right before S_1 is in a domain, a point between S_1 and S_2 is out of the domain and a point right after S_2 is again in the domain. Since only an S point crosses a border line, no such crossing can occur between two consecutive P points, which therefore can only be of the same kind. When two P points are separated by an odd number of S points, they can only be of a different kind. If separated by an even number of S points they can only be of the same kind.

III. The union of two conjoint curves consists of all the outer points (the $\underline{P}$'s) of both curves and all their intersection points (the S's), properly ordered.

IV. Their intersection consists of all the inner points (the $\overline{P}$'s) of both curves and all their S points properly ordered.

V. The difference of two conjoint curves consists of all the outer points of the substrahend curve, all the inner points of the substracter curve and all their S points.

The above properties suggest the following steps for our algorithm:

(1). derive the intersection points of the curves and expand their point vectors to include them properly inserted (discussed in section 4).

(2). characterize the expanded vectors derived as above by distinguishing its P points into $\underline{P}$ and $\overline{P}$ (discussed in section 5).

(3). properly mix and thread the points in the characterized expansions to derive the union, the intersection or a difference (section 6).

4. Expansions

Given two conjoint curves C^a and C^b, we denote C^{ab} the expansion of C^a with respect to C^b. The expansion of C^a with respect to

C^b is derived by properly inserting in C^a's point vector the points at which C^a and C^b intersect. Similarly, C^{ba} is the expansion of C^b with respect to C^a. For example, if C^a and C^b are as in Figure 2.5, that is

$$C^a = (P_0^a, P_1^a, P_2^a, P_3^a, P_4^a, P_0^a) \quad \text{and} \quad C^b = (P_0^b, P_1^b, P_2^b, P_3^b, P_0^b) \ ,$$

then $C^{ab} = (P_0^a, S_2, S_1, P_1^a, S_3, P_2^a, S_4, P_3^a, P_4^a, S_5, S_6 P_0^a)$

and $C^{ba} = (P_0^b, P_1^b, S_4, S_3, S_1, S_5, P_2^b, S_6, S_2, P_3^b, P_0^b)$.

The conjunction of C^a and C^b is regular.

To depict the points of intersection, it is necessary to sequentially take each and every line segment of the one curve C^k with each and every line segment of the other curve $C^{k'}$. Each such pair of line segments (one from each curve) is tested for an intersection by applying the function INS. which is defined as follows:

$$\begin{aligned} \text{INS } (P_{i_1}^a, P_{i_2}^a, P_{i_1'}^b, P_{i_2'}^b) &= (S,S) \quad \text{if the segments intersect} \\ &= (\emptyset,\emptyset) \quad \text{otherwise.} \end{aligned}$$

The precise definition of INS is given in Appendix I.

We have written the output of INS as a pair of points to indicate that each time a point of intersection is depicted, it is inserted in the point vectors of both curves. The depicted S point is inserted between the P points delineating the intersecting line segments. To assure that consecutive S points are properly ordered, each time an S point is depicted it breaks the segment to which it belongs into two portions and each is tested independently. For example, assume that (P_1, P_2) is intersected at two points S_1 and S_2 and that S_1 is depicted first. Then (P_1,P_2) becomes (P_1,S_1,P_2), and is interpreted and tested as two line segments; (P_1,S_1) and (S_1,P_2). If the second time (P_1,P_2) is intersected before S_1, then the tests for (P_1,S_1) will generate S_2 and the tests for (S_1,P_2) will generate $\emptyset$. The result will be (P_1,S_2,S_1,P_2). In further tests (P_1,S_2,S_1,P_2) will be treated as three segments.

To cover the irregular cases, the above operation is slightly expanded. When the curves intersect at corner points, rather than inserting a new S point, we change an existing P point into an S point. We proceed as follows.

The segments to be checked are accompanied by their next segments also. That is if we want to check (P^a_1,P^a_2) and (P^b_1,P^b_2) for intersection we take (P^a_1,P^a_2,P^a_3) and (P^b_1,P^b_2,P^b_3). If (P^a_1,P^a_2), (P^b_1,P^b_2) is a non-intersecting pair, we do nothing and proceed with the pair to be checked next. If it is an intersecting pair, we insert the depicted S point in both segments before we proceed with the next test. If it is a tangent pair, we also need to consider P^a_3 and P^b_3 (either or both depending on whether it is a single or a double sided tangency). The different cases of tangent pairs are summarized in Figure 4.1.

<u>Figure 4.1 about here</u>

In columns 1 and 2 (Fig. 4.1) we have double sided tangencies, and column 1 differs from 2 only by the direction of the segments. In columns 3 and 4 the tangencies are single sided and differ as above. In the cases of row 1, the tangent point is an intersection point. In 1.1 and 1.2 no new S point is inserted, but the existing points P^a_2 and P^b_2 are changed into S points. We indicate this by adding a superscript s to the original points. In 1.3 and 1.4 one new S point is inserted and one existing P point is changed into an S point. In the cases of rows 2, 3 and 4, the tangent points are not considered intersection points. Even though these points are boundary points for both domains, semantically they work as outer or inner and need to be characterized. Their character depends on the directions and relative positions of the segments involved and are as shown in Figure 4.1. As before, we write $\underline{P}$ for an outer and $\overline{P}$ for an inner P point. In cases 2.3, 2.4, 3.3 and 3.4 one new P point, properly characterized, is inserted in one of the curves. The inserting of a new point is denoted with a prime. In all other cases no new point is inserted, but existing P points are properly characterized. In cases 4.1 and 4.2 there is no direct way to characterize the tangent points. In these cases and for both curves, we leave the tangent P points temporarily with no characterization; they will be characterized later by the second part of the algorithm. For the above checks we employ the function INT which is a generalized version of INS and its precise definition is given in Appendix I.

To summarize

$$\mathrm{INT}\ (P^a_i, P^a_{i+1}, P^a_{i+2}, P^b_j, P^b_{j+1}, P^b_{j+2})$$

$= (\emptyset,\emptyset)$ if the pair of segments is non-intersecting,

$= (S,S)$ if the pair is intersecting,

$= (P^s,P^s)$ if tangent and case 1.1 or 1.2,

$= (S,P^s)$ or (P^s,S) if tangent and case 1.3 or 1.4,

$= (\overline{P},\underline{P})$ or $(\underline{P},\overline{P})$ if tangent and case 2.1 or 3.1,

$= (\underline{P},\underline{P})$ if tangent and case 2.2 or 3.2,

$= (\overline{P}',\underline{P})$ or $(\overline{P},\underline{P}')$ or $(\underline{P}',\overline{P})$ or $(\underline{P},\overline{P}')$ if case 2.3 or 3.3,

$= (\underline{P}',\underline{P})$ or $(\underline{P},\underline{P}')$ if tangent and case 2.4 or 3.4,

$= (P,P)$ if tangent and case 4.1 or 4.2.

The notation P^s is used above to indicate that the respective P point is changed into an S and a prime ($\underline{P}'$ or $\overline{P}'$) to indicate that the respective P point is a new insertion.

As an example consider the irregular pair of conjoint curves shown in Figure 4.2. They are

$$C^c = (P_0^c,P_1^c,P_2^c,P_3^c,P_4^c,P_5^c,P_6^c,P_7^c,P_0^c) \quad \text{and} \quad C^d = (P_0^d,P_1^d,P_2^d,P_3^d,P_4^d,P_5^d,P_0^d).$$

Their expansions will be derived as

$$C^{cd} = (P_0^c,S_1,\overline{P}_1^c,\underline{P}_2^c,P_3^c,\underline{P}_2^d,P_{43}^{cd},P_{54}^{cd},P_{65}^{cd},\overline{P}_7^c,S_2,P_0^c) \quad \text{and}$$

$$C^{dc} = (P_0^d,S_2,S_1,P_1^d,\underline{P}_1^c,\overline{P}_2^c,\overline{P}_2^d,P_{43}^{cd},P_{54}^{cd},P_{65}^{cd},\underline{P}_7^c,P_0^d) .$$

When two p-points coincide, the notation denotes their dual definition, e.g., p_{48}^{cd}.

Figure 4.2 about here

5. Characterization of the Expansions

An expansion is characterized when its P points are distinguished into $\underline{P}$ (outer) and $\overline{P}$ (inner) points. To characterize an expansion from a regular conjunction, it suffices to know the character of any one of its P points, by property II (section 3). With such a point as a basis, the remaining P-points are sequentially traced and characterized. Each next P-point's character depends on the previous and the number of interfering S points. If none or an even number of S points interfere, the P point takes the character of its previous P point. It takes the reverse character otherwise (that is, if an odd number of S points intervene). For example, given the expansions C^{ab} and C^{ba} as derived in section 4 and with their closing points characterized, that is

$$C^{ab} = (\underline{P}_0^a, S_2, S_1, P_1^a, S_3, P_2^a, S_4, P_3^a, P_4^a, S_5, S_6, \underline{P}_0^a)$$

and $$C^{bc} = (\underline{P}_0^b, P_1^b, S_4, S_3, S_1, S_5, P_2^b, S_6, S_2, P_3^b, \underline{P}_0^b)$$

then the whole expansions are characterized as

$$C^{ab'} = (\underline{P}^a{}_0, S_2, S_1, \underline{P}_1^a, S_3, \bar{P}{}^a_2, S_4, \underline{P}_3^a, \underline{P}_4^a, S_5, S_6, \underline{P}_0^a)$$

and $$C^{ba'} = (\underline{P}_0^b, \underline{P}_1^b, S_4, S_3, S_1, S_5, \underline{P}_2^b, S_6, S_2, \underline{P}_3^b, \underline{P}_0^b) .$$

The primes in $C^{ab'}$ and $C^{ba'}$ indicate that the respective expansions are characterized. The above results can be verified with Figure 2.5.

At this point, we introduce the restriction that the point vectors of the domains under consideration should be given in a normal form. A point vector is in a normal form when its closing point is unique (it does not belong to the other curve also) and it is characterized. If in a certain problem area it is not realistic or practical to assume that the point vectors under consideration are in normal forms, an arbitrary point vector can be normalized by the normalizer, which is a fairly simple mechanical procedure and is defined in Appendix II. The only case where a point vector cannot be normalized is when it refers to a coincident pair of domains.

Given the initial point vectors in normal form, the derived expansions will be in normal form also, and therefore they will have their closing points characterized. Then for curves from regular and irregular conjunctions alike, we can start at the closing point and sequentially characterize the remaining P points. What is different between the regular and the irregular case is that in the second, the expanded vector contains some more P points (in addition to its closing point) which are already characterized. These points are non-intersecting tangent points and have been characterized by the first part of the algorithm, when the expansions were derived. Consequently, as the sequential characterization of the P points proceeds, it is possible to encounter a case where two consecutive P points are not of the same character. This contradicts property II and implies that some S point is missing. We call such pairs of consecutive P points non-conforming pairs, and when depicted, we change the second P point into an S. Given two conjoint curves, their normalized expansions can only have the same number of non-conforming pairs.

For example, assuming that the vectors for C^c and C^d (section 4, Figure 4.2) were given in normal forms, their expansions would have been

$$C^{cd} = (\underline{P}_0^c, S_1, \overline{P}_1^c, \underline{P}_2^c, P_3^c\ \underline{P}_2^d, P_{43}^{cd}, P_{54}^{cd}, P_{65}^{cd}, \overline{P}_7^c, S_2, \underline{P}_0^c)$$

$$\text{and}\quad C^{dc} = (\underline{P}_0^d, S_2, S_1, P_1^d, \underline{P}_1^c, \overline{P}_2^c, \overline{P}_2^d,\ P_{43}^{cd}, P_{54}^{cd}, P_{65}^{cd}, \underline{P}_7^c, \underline{P}_0^d)\ .$$

There is at least one obvious non-conforming pair in each of the above expansions; $(\overline{P}_1^c, \underline{P}_2^c)$ in C^{cd} and $(\underline{P}_1^c, \overline{P}_2^c)$ in C^{dc}. They are changed into $(\overline{P}_1^c, S_2^c)$ and $(\underline{P}_1^c, S_2^c)$ respectively. As the characterization of the other p-points proceeds, one more non-conforming pair is depicted in each expansion. They are $(\underline{P}_{65}^{cd}, \overline{P}_7^c)$ in C^{cd} and $(\overline{P}_{65}^{cd}, \underline{P}_7^c)$ in C^{dc}. They are changed into $(\underline{P}_{65}^{cd}, S_7^c)$ and $(\overline{P}_{65}^{cd}, S_7^c)$. The characterized expansions will be

$$C^{cd'} = (\underline{P}_0^c, S_1, \overline{P}_1^c, S_2^c, \underline{P}_3^c, \underline{P}_2^d, \underline{P}_{43}^{cd}, \underline{P}_{54}^{cd}, \underline{P}_{65}^{cd}, S_7^c, S_2, \underline{P}_0^c)$$

and $$C^{dc'} = (\underline{P}_0^d, S_2, S_1, \underline{P}_1^d, \underline{P}_1^d, S_2^c, \overline{P}_2^d, \overline{P}_{43}^{cd}, \overline{P}_{54}^{cd}, \overline{P}_{65}^{cd}, S_1^c, \underline{P}_0^d)$$

6. The Threading Algorithm

Given the expanded and characterized point vectors of two conjoint curves, by properly mixing and threading their points, we can derive the curve sets that delineate their union, intersection and differences. In each case, the guidelines for the proper selection and threading of the points are given by properties III, IV and V of section 3. The same threading algorithm applies to all cases. What differs is the definition of the starting point B and the order by which the point vectors under consideration are written.

For the union and the intersection both vectors are written in their original order. For the differences, the subtrahend vector is written in its positive order and the subtracter in negative order. For example, to derive $C^a - C^b$, $C^{ba'}$ is written in reverse order.

The starting point B is always an intersection (S) point and can be in either one of the two expansions. For the union, it is any S point followed by a $\underline{P}$ (outer) point or preceded by a $\overline{P}$ (inner) point. For the intersection, B is any S followed by a $\overline{P}$ or preceded by a $\underline{P}$. For the differences, if B is in the subtrahend expansion, it is defined as for the union; if it is in the subtracter expansion as for the intersection. The definitions of B for the different cases are summarized in Figure 6.1(a). The B point is indicated by a circled S. Using 1 for ($S\,\underline{P}$ or $\overline{P}\,S$) and 0 for ($S\,\overline{P}$ or $\underline{P}\,S$), the different cases can be coded by the binary numbers as shown in Figure 6.1(b). This coding will be used to instruct the system which operation is desired (sections 7 and 8).

<u>Figure 6.1 about here</u>

The S points in the expansions delineate strings of either $\underline{P}$ or $\overline{P}$ points. In addition, because the S points are common in both expansions, they link a string of P points of the one expansion with a string of P points of the other. The form of linkage can be specified. By definition, if an S point has a P point on each side, these P points can only be of a different kind. Furthermore (and assuming that the two expansions have been written in the same order), if in one expansion an S point is followed by a $\underline{P}$ point (or/and preceded by a $\overline{P}$ point), the situation is reversed in the other expansion; that is, the same S point is followed by a $\overline{P}$ point (or/and preceded by a $\underline{P}$ point). Consequently, if we start at an S point in the one expansion and positively trace and thread all the P points on its right till we reach the next S point; then transfer to the identical S point of the other expansion and do the same, and keep on transferring from one expansion to the other and stop when we come back to the S point we started with, we pick P points of only one character. Which character depends upon the S point we started with. If it is a union B point, we pick all the $\underline{P}$ points, as required by property III; if it is an intersection B point, we pick all the $\overline{P}$ points, as required by property IV.

For the differences we do not want to pick P points of one kind. Instead, we want to pick the $\underline{P}$ points from the subtrahend expansion and the $\overline{P}$ points from the subtracter. If we reverse the order of the subtracter expansion, then if an S point is followed by a $\underline{P}$ point in the one expansion, so it is in the other also, and vice versa. Consequently, by starting at the appropriate S point and tracing the expansions as before, we end up with a proper mixture of $\overline{P}$ and $\underline{P}$ points as is required for the differences.

To summarize, the threading algorithm operates as follows:

1. write the expansions properly and as required in each case;
2. identify the starting point B, its definition depending on the case;
3. start at B, thread it, move to the right threading all P points up to the next S point. Delete it and jump to the identical S point (call it S') of the other expansion;
4. start at S', thread this and all the P points to the right, up to the next S point. If this is not the same with B, delete it, jump to S' and repeat step 4. If S=B, thread it and go to step 5;
5. one curve of the curve set has been derived. Identify the next B, if one exists, and go to step 3. If no more S points are left in the expansion, STOP.

In Figure 6.2, we show the derivation of $C^a \cup C^b$. We derived
$C^a \cup C^b = \{(S_1,P_1^a,S_3,S_1),(S_4,P_3^a,P_4^a,S_5,P_2^b,S_6,P_0^a,S_2,P_3^b,P_0^b,P_1^b,S_4)\}$.
In a similar fashion we derive

$$C^a \cap C^b = \{(S_3,P_2^a,S_4,S_3),(S_5,S_6,S_2,S_1,S_5)\},$$

$$C^a - C^b = \{(S_1,P_1^a,S_3,S_4,P_3^a,P_4^a,S_5,S_1),(S_6,P_0^a,S_2,S_6)\} \quad \text{and}$$

$$C^b - C^a = \{(S_1,S_2,P_3^b,P_0^b,P_1^b,S_4,P_2^a,S_3,S_1),(S_5,P_2^b,S_6,S_5)\}.$$

<u>Figure 6.2 about here</u>

For the derivation of the differences of irregular junctions we shall add one more step to the threading algorithm. The curve sets derived for the differences may contain a curve, some portion of which is tracing back and forth the same sequence of line segments and delineates no area. This case occurs when two curves have tangent line segments. For example, in Figure 6.3, we show the derivation of $C^c - C^d$. We derive

$$C^c - C^d = \{(S_2^c, P_3^c, P_2^d, P_{43}^{cd}, P_{54}^{cd}, P_{65}^{cd}, S_7^c, P_{65}^{cd}, P_{54}^{cd}, P_{43}^{cd}, P_2^d, S_2^c), (S_2, P_0^c, S_1, S_2)\}.$$

Notice that only one portion of the first curve delineates an area. This is $(S_2^c, P_3^c, P_2, S_2^c)$. The remaining traces back and forth the segments $(P^d{}_2, P_{43}^{cd}, P_{54}^{cd}, P_{65}^{cd}, S_7^c)$. To correct such situations, the curve sets derived for the differences are examined and the segments that delineate no area are eliminated. Applying this step, the above curve set becomes

$$C^c - C^d = \{(S_2^c, P_3^c, P_2^d, S_2^c), (S_2, P_0^c, S_1, S_2)\}.$$

Figure 6.3 about here

7. The Bead Machine

The derivation of the union, intersection or a difference of two conjoint domains can be precisely defined in terms of the mechanism shown in Figure 7.1. We shall be calling it the Bead Machine and referring to it as M^b. The name reflects the analogy that the points of the curves under consideration are written on beads. This emphasizes the fact that their bulk is significant and when they are pushed into some location, the point (bead) already there is not lost but is pushed further up or down. The beads are of different colors; the P beads are red, the S beads are grey, the $\overline{P}$ beads are white and the $\underline{P}$ beads are black. Thus, a bead depicts three variables, two that define the x and y coordinates of a point and one defining a color.

Figure 7.1 about here

M^b consists of a central processing unit (CU), and R1 and R2 which are two ring-like double-ended stacks, holding the point vectors (beads) of the domains under consideration. RO_1 and RO_2 are two registers in CU. H1 and H2 are two linear stacks used for garbage disposal, and W is again a linear stack functioning as an output buffer.

The indices of R1, R2, H1, H2 and W are $i_1 = 1, \ldots, n_1$, $i_2 = 1, \ldots, n_2$, $j_1 = 1, \ldots, m_1$, $j_2 = 1, \ldots, m_2$ and $k = 1, \ldots, v$ respectively. The upper limits n_1, n_2, m_1, m_2 and v are variables and depend on the specific pair of domains under consideration. Therefore R1, R2, H1, H2 and W are of variable lengths and expandable. That is the CU has the capability of increasing and decreasing the values of the upper limits and consequently of adding and subtracting cells from R1, R2, H1, H2 and W.

As we did with the algorithm, we shall divide M^b into three parts to be referred as M^{b1}, M^{b2} and M^{b3}. M^{b1} expands the initial point vectors. It starts with the vectors stored in R1 and R2 and ends with their expansions stored in the same stacks. M^{b2} characterizes the expansions. It is given the expansions, as derived by M^{b1}, stored in R1 and R2 and stores the characterized expansions in the same stacks. M^{b3} is given the characterized expansions (stored in R1 and R2) and threads the curve set for the union, the intersection or one of the differences. It stores the results in W. Which curve it derives depends on the instructions given to it. After it derives the curve set for one of the operations, it can proceed with the derivation of another, if the characterized expansions are re-entered in R1 and R2 and M^{b3} is given further instructions.

When the initial vectors or their characterized expansions are entered in R1 and R2, their closing points are required to occupy locations $R1_1$, $R1_{n_1}$ and $R2_1$, $R2_{n_2}$. In addition to RO_1 and RO_2, $R1_{n_1}$, $R1_1$, $R1_2$, $R2_{n_2}$, $R2_1$, $R2_2$, and the whole W are directly accessible by CU.

Formally, a Bead Machine M^b is a system (L,K,T,G,Q,δ), where L is the hardware shown in Figure 7.1, K a set of computations, T a set of transfers, G a set of colorings, Q a set of states and δ a mapping of Q*L into Q*K*G*T.

Specifically:

$$K = \{k^1: \ (RO_1, RO_2) \leftarrow INT\ (R1_{n_1}, R1_1, R1_2, R2_{n_2}, R2_1, R2_2)\}$$

$$T = \{t^1: \ R1_{i_1} \leftarrow R1_{i_1+1} \quad (i_1=1,\ldots, n_1 \ (\mathrm{mod}\ n_1))$$

$$t^2: \ R2_{i_2} \leftarrow R2_{i_2+1} \quad (i_2 = 1,\ldots,n_2 (\mathrm{mod}\ n_2))$$

$$t^3: \ n_1 = n_1 + 1;\ R1_{i_1+1} \leftarrow R1_{i_1} \quad (i_1 = 1,\ldots,n_1 - 1);\ R1_1 \leftarrow RO_1$$

$$t^4: \ n_2 = n_2 + 1;\ R2_{n_2} \leftarrow RO_2$$

$$t^5: \ H1_{m_1} \leftarrow R1_1;\ R1_1 \leftarrow RO_1;\ m_1 = m_1 + 1$$

$$t^6: \ H2_{m_2} \leftarrow R2_1;\ R2_1 \leftarrow RO_2;\ m_2 = m_2 + 1$$

$$t^7: \ k = k+1;\ W_k \leftarrow R1_{n_1};\ n_1 = n_1 - 1;\ t^1$$

$$t^8: \ k = k+1;\ W_k \leftarrow R2_{n_2};\ n_2 = n_2 - 1;\ t^2$$

t^9: $H1_{m_1} \leftarrow R1_{n_1}$; $m_1 = m_1 + 1$; $n_1 = n_1 - 1$; t^1

t^{10}: $H2_{m_2} \leftarrow R2_{n_2}$; $m_2 = m_2 + 1$; $n_2 = n_2 - 1$; t^2

t^{11}: $k=0$; $m_1 = 1$; $m_2 = 1$

t^{12}: $g = k$ $\}$

$$G = \{c^1: R1_1 \overset{\mathcal{E}}{\leftarrow} \overline{P}\ ,\ C^2: R2_1 \overset{\mathcal{E}}{\leftarrow} \overline{P}\ ,$$

$$c^3: R1_1 \overset{\mathcal{E}}{\leftarrow} \underline{P}\ ,\ C^4: R2_1 \overset{\mathcal{E}}{\leftarrow} \underline{P}\ ,$$

$$c^5: R1_1 \overset{\mathcal{E}}{\leftarrow} S\ ,\ C^6: R2_1 \overset{\mathcal{E}}{\leftarrow} S\ \}$$

$$Q = \{q^o, q^i\ (i=1,\ldots,), q^f\}$$

In G, the symbol $\overset{\mathcal{E}}{\leftarrow}$ indicates that the bead in the left hand side location is colored as the right hand side indicates, where $\overline{P}$ is black, $\underline{P}$ is white and S is grey.

The mappings constitute M^b's program, so to speak. $\delta(a) = (b)$ means given (a), where (a) a state and a set of relations over the content of CU's registers, do (b), where (b) is a change of state and (possibly empty) set of computations, coloring and/or transfers. For the relations of the registers we use three types of equivalence symbols and their negations; $=$(and $\neq$) refers to the coordinates of the points; $\equiv$(and $\not\equiv$) refers to the colors; and $\simeq$(and $\not\simeq$) refers to both the colors and the coordinates. We shall separately list and shortly discuss the mappings of the three parts of M^b (that is of M^{b1}, M^{b2} and M^{b3}). The index in the parentheses (far right) indicates how they can be put together.

For M^{b1} the mapping δ is as follows:

$$\delta\ (q^o,\ \{R2_{n_2} = R2_1\}) = (q^3, t^1) \tag{1}$$

$$\delta\ (q^o,\ \{(R1_{n_1} \simeq R2_{n_2} \not\equiv P) \vee (R1_{n_1} \simeq R2_1 \not\equiv P) \vee (R1_1 \simeq R2_{n_2} \not\equiv P) \vee (R1_1 \simeq R2_1 \not\equiv P)\}) = (q^o, t^2) \tag{2}$$

$$\delta\ (q^o,\ \{R2_{n_2} \neq R2_1\}) = (q^1, k^1) \tag{3}$$

$$\delta\ (q^1,\ \{RO_1 = RO_2 = \emptyset\}) = (q^o, t^2) \tag{4}$$

$$\delta\ (q^1,\ \{RO_1 \neq R1_1\}) = (q^2, t^3) \tag{5}$$

$$\delta\ (q^1,\ \{RO_1 = R1_1\}) = (q^2, t^5) \tag{6}$$

$$\delta\ (q^2,\ \{RO_2 \neq R2_1\}) = (q^o, t^4, t^2) \tag{7}$$

$$\delta\ (q^2,\ \{RO_2 = R2_1\}) = (q^o, t^6, t^2, t^2) \tag{8}$$

$$\delta\ (q^3,\ \{R1_{n_1} \not\simeq R1_1\}) = (q^o, t^2) \tag{9}$$

$$\delta\ (q^3,\ \{R1_{n_1} \simeq R1_1 \equiv \overline{P}\}) = (q^4, t^1, t^{11}) \tag{10}$$

$$\delta\ (q^3,\ \{R1_{n_1} \simeq R1_1 \equiv \underline{P}\}) = (q^5, t^1, t^{11}) \tag{11}$$

When M^{b1} starts, the point vectors of the domains under consideration are in normal forms and already stored in R1 and R2, with their closing points in $R1_1$, $R1_{n_1}$ and $R2_1$, $R2_{n_2}$ respectively. The initial state of M^{b1} is q^o and its final states are q^4 and q^5. At q^o, M^{b1} applies k^1 which computes the points of intersection (when they exist) or characterizes the tangent points (map. 3). The results are stored in

RO_1 and RO_2 and M^{b1} goes to states q^1 and q^2 which check and properly inserts them in R1 and R2 (map. 4-8). Then it returns to q^o which proceeds with the next pair of segments. Before doing so, it checks if the segments under consideration have a common S point, in which case it skips this pair and takes the next (map. 2). Mapping 1 checks if R2 has completed a cycle, in which case it goes to state q^3 and moves the beads in R1 down by one position (map. 9), unless R1 has also completed a cycle. In the latter case the process is completed and M^{b1} goes to either one of its final states. It goes to q^4 if the closing point in $R1_1$ is $\overline{P}$ (map. 10) or it goes to q^5 if it is $\underline{P}$ (map. 11).

For M^{b2} the mapping o is as follows:

$$\delta\ (q^4,\ \{R1_2 \simeq\ R1_1 \wedge R2_1 \equiv \overline{P}\}) = (q^6 t^1, t^2) \tag{12}$$

$$\delta\ (q^4,\ \{R1_2 \simeq\ R1_1 \wedge R2_1 \equiv \underline{P}\ \}) = (q^7 t^1, t^2) \tag{13}$$

$$\delta\ (q^4,\ \{R1_1 \equiv P\}) = (q^4, c^1, t^1) \tag{14}$$

$$\delta\ (q^4,\ \{R1_1 \equiv S \vee R1_1 \equiv \underline{P}\}) = (q^5, c^5, t^1) \tag{15}$$

$$\delta\ (q^4) = (q^4, t^1) \tag{16}$$

$$\delta\ (q^5,\ \{R1_2 \simeq\ R1_1 \wedge R2_1 \equiv \overline{P}\}) = (q^6 t^1, t^2) \tag{17}$$

$$\delta\ (q^5,\ \{R1_2 \simeq\ R1_1 \wedge R2_1 \equiv \underline{P}\ \}) = (q^7 t^1, t^2) \tag{18}$$

$$\delta\ (q^5,\ \{R1_1 \equiv P\}) = (q^5, c^3, t^1) \tag{19}$$

$$\delta\ (q^5, \{R1_1 \equiv S \vee R1_1 \equiv \overline{P}\}) = (q^4, c^5, t^1) \tag{20}$$

$$\delta\ (q^5) = (q^5, t^1) \tag{21}$$

$$\delta\ (q^6,\ \{R2_2 \simeq\ R2_1\}) = (q^{10} t^2) \tag{22}$$

$$\delta\ (q^6,\ \{R2_1 \equiv P\}) = (q^6, c^2, t^2) \tag{23}$$

$$\delta\ (q^6,\ \{R2_1 \equiv S \vee R2_1 \equiv \underline{P}\}) = (q^7, c^6, t^2) \tag{24}$$

$$\delta\ (q^6,\ \{\text{otherwise}\}) = (q^6, t^2) \tag{25}$$

$$\delta\ (q^7,\ \{R2_2 \simeq R2_1\}) = (q^{10}) \tag{26}$$

$$\delta\ (q^7,\ \{R2_1 \equiv P\}) = (q^7, c^4, t^2) \tag{27}$$

$$\delta\ (q^7,\ \{R2_1 \equiv S \vee R2_1 \equiv \overline{P}\}) = (q^6, c^6, t^2) \tag{28}$$

$$\delta\ (q^7,\ \{\text{otherwise}\}) = (q^9, t^2) \tag{29}$$

M^{b2} characterizes the points in each of the curves. It consists of two sections; the first (map. 12-21) characterizes the points in R1 and the second (map. 22-29) in R2. The final state of M^{b2} is q^{10}. (The missing mappings 30-34 are described in section 8.)

For M^{b3}, the mapping δ is as follows:

$$\delta\ (q^{10}, \{\}) = (\langle\text{instruction}\rangle,\ q^{11}, t^9, t^{10}, t^{11}) \tag{35}$$

$$\delta\ (q^{11},\ \{RO_1 = RO_2 = \emptyset\}) = (q^f) \tag{36}$$

$$\delta\ (q^{11},\ \{R1_{n_1} \simeq R1_1 \simeq R2_{n_2} \simeq R2_1\}) = (q^{10}) \tag{37}$$

$$\delta\ (q^{11},\ \{(RO_1 = 0 \wedge R1_{n_1} \equiv S \wedge R1_1 \equiv \underline{P}) \vee (RO_1 = 1 \wedge R1_{n_1} \equiv \tag{38}$$

$$S \wedge R1_1 \equiv \overline{P})\}) = (q^{13}, t^7, t^{12}) \tag{39}$$

$$\delta\ (q^{11},\ \{(RO_1 = 0 \wedge R1_{n_1} \equiv \overline{P} \wedge R1_1 \equiv S) \vee (RO_1 = 1 \wedge R1_{n_1} \equiv$$

$$\underline{P} \wedge R1_1 \equiv S)\}) = (q^{13}, t^9, t^7, t^{12}) \tag{40}$$

$$\delta\ (q^{11},\ \{(RO_1 = 0 \wedge R1_{n_1} \equiv \overline{P} \wedge R1_1 \equiv \overline{P}) \vee (RO_1 = 1 \wedge R1_{n_1} \equiv$$

$$\underline{P} \wedge R1_1 \equiv \underline{P})\}) = (q^{11}, t^9)$$

$$\delta\ (q^{11},\ \{(RO_2 = 0 \wedge R2_{n_2} \equiv S \wedge R2_1 \equiv \underline{P}) \vee (RO_2 = 1 \wedge R2_{n_2} \equiv S \wedge R2_1 \equiv \overline{P})\}) = (q^{15}, t^8, t^{12}) \tag{41}$$

$$\delta\ (q^{11},\ \{(RO_2 = 0 \wedge R2_{n_2} \equiv \overline{P} \wedge R2_1 \equiv S) \vee (RO_2 = 1 \wedge R2_{n_2} \equiv \underline{P} \wedge R2_1 \equiv S)\}) = (q^{15}, t^{10}, t^8, t^{12}) \tag{42}$$

$$\delta\ (q^{11},\ \{(RO_2 = 0 \wedge R2_n2 \equiv \overline{P} \wedge R2_1 \equiv \overline{P}) \vee (RO_2 = 1 \wedge R2_{n_2} \equiv \underline{P} \wedge R2_1 \equiv \underline{P})\}) = (q^{11}, t^{10}) \tag{43}$$

$$\delta\ (q^{11},\ \{\text{otherwise}\}) = (q^{11}, t^1, t^2) \tag{44}$$

$$\delta\ (q^{12},\ \{R2_{n_2} \simeq W_g\}) = (q^{11}, t^8) \tag{45}$$

$$\delta\ (q^{12},\ \{R2_{n_2} \not\simeq W_g \wedge R2_{n_2} \not\simeq R1_{n_1}\}) = (q^{12}, t^1) \tag{46}$$

$$\delta\ (q^{12},\ \{R2_{n_2} \not\simeq W_g \wedge R2_{n_2} \simeq R1_{n_1}\}) = (q^{13}, t^8, t^9) \tag{47}$$

$$\delta\ (q^{13},\ \{R1_{n_1} \not\equiv S\}) = (q^{13}, t^7) \tag{48}$$

$$\delta\ (q^{13},\ \{R1_{n_1} \equiv S\}) = (q^{14}) \tag{49}$$

$$\delta\ (q^{14},\ \{R1_{n_1} \simeq W_g\}) = (q^{11}, t^7) \tag{50}$$

$$\delta\ (q^{14},\ \{R1_{n_1} \not\simeq W_g \wedge R1_{n_1} \not\simeq R2_{n_2}\}) = (q^{14}, t^2) \tag{51}$$

$$\delta\ (q^{14},\ \{R1_{n_1} \not\simeq W_g \wedge R1_{n_1} \simeq R2_{n_2}\}) = (q^{15}, t^7, t^{10}) \tag{52}$$

$$\delta\ (q^{15},\ \{R2_{n_2} \not\equiv S\}) = (q^{15}, t^8) \tag{53}$$

$$\delta\ (q^{15},\ \{R2_{n_2} \equiv S\}) = (q^{12}) \tag{54}$$

The initial state of M^{b3} is q^{10}. At q^{10}, instructions with respect to which curve set is desired should be given (map. 35). The instructions are stored in RO_1 and RO_2, in binary form. The instructions are of the form 00 for the union, 11 for the intersection 01 and 10 for the differences. See Figure 6.1. M^{b3} then goes to state q^{11}. At q^{11}, it picks an appropriate starting point B as defined by the instructions and goes to state q^{13} or q^{15}, depending on whether B is in R1 or in R2 (map. 38-42). At states q^{13} and q^{15} it threads all the P ($\overline{P}$ or $\underline{P}$) points up to the next S. At each S it goes to q^{14} or q^{12} to check if S B (map. 48-50 and 43-45). For this check it compares the newly encountered S with the content of W_g, where B is stored. If the check is negative, it goes to state q^{13} or q^{15} and proceeds up to the next S. If the test is positive, it has completed the threading of one curve of the curve set and goes to state q^{11} ready to search for the next curve. If R1 and R2 are empty (map. 37), the whole curve set has been derived, and it goes to q^{10} for further instructions. If no more instructions are given and $RO_1 = RO_2 = \emptyset$, the job is done and M^b goes to its final state q^f (map. 36).

We have made the assumption that the point vectors of the domains under consideration, when initially entered into R1 and R2, are in normal forms. As already pointed out, in some problem areas, such an assumption is not realistic or practical. In these cases, the point vectors can be normalized by the use of the normalizer M^{b0}, which is described in Appendix II. Also, in Appendix I, we give the precise definition of the function INT which constitutes k^I and which is extensively utilized by M^{b1}. Finally, the curve segments for the differences of irregular curves with tangent segments, as derived by M^b may contain portions which trace back and forth a sequence of segments and delineate no area. M^{b1} or M^{b2} or M^{b3} can be expanded to provide for the direct elimination of such cases. Alternatively, an a posteriori test applied to the difference curves only may eliminate the redundant points. For conciseness, we omit this editing operation and our development assumes the latter method of eliminating the redundant points.

8. M^b for Disjoint and Subjoint Domains

M^b, as defined in section 7, is not quite general. It does not work if two domains are not conjoint. To generalize M^b, the mappings 21 and 26 will be changed and five new mappings will be added to M^{b2}, as follows:

$$\delta\ (q^6,\ \{R2_2 \simeq R2_1\}) = (q^8, t^1, t^2) \tag{21}$$

$$\delta\ (q^7,\ \{R2_2 \simeq R2_1\}) = (q^8, t^1, t^2) \tag{26}$$

$$\delta\ (q^8,\ \{R1_{n_1} \simeq R1_1\}) = (q^{20}) \tag{30}$$

$$\delta\ (q^8,\ \{R1_{n_1} \not\simeq R1_1 \wedge R1_1 \not\equiv S\}) = (q^8, t^1) \tag{31}$$

$$\delta\ (q^8,\ \{R1_{n_1} \not\simeq R1_1 \wedge R1_1 \equiv S\}) = (q^9, t^1) \tag{32}$$

$$\delta\ (q^9,\ \{R1_{n_1} \simeq R1_1\}) = (q^{10}) \tag{33}$$

$$\delta\ (q^9,\ \{R1_{n_1} \not\simeq R1_1\}) = (q^9, t^1) \tag{34}$$

Now, by mapping 21 and 26, M^{b2} does not go directly to q^{10}, but to q^8. At q^8 (map. 30-32), it checks if an S point has been depicted. If not, it goes to q^{20}. If yes, it goes to q^9 where it resets R1 and then goes to q^{10} (map. 33-34). If an S point is found, the domains are conjoint and q^{10} is the initial state of M^{b3}, which is as before. If an S point is not found, the curves do not intersect and, given that they are not coincident, they are disjoint, if the closing points of both are $\underline{P}$ (outer), or subjoint, if at least one of the closing points is $\overline{P}$ (inner). In such cases M^{b2} goes to q^{20} which is the initial state of M^{b4}, the new part we shall add to M^{b}. Our assumption that the two domains are not coincident is based on our earlier assumption that the point vectors were given in normal forms. For coincident pairs of domains no normal forms are derivable. If a normalizer is used before the main parts of M^{b}, it depicts the coincident pairs, as discussed in Appendix II.

$M^{b4} = (H,K,T,G,Q,\theta)$ where H as before, K and G are empty.

$$T = \{t^{21}\colon W_{k+i_1} \leftarrow R1_{i_1}\ (i_1 = 1,\ldots,n_1)\ ;\ k = k+n_1$$

$$t^{22}\colon\ W_{k+i_2} \leftarrow R2_{i_2} (i_2 = 1,\ldots,n_2)\ :\ k = k+n_2$$

$$t^{23}\colon\ W_{k+n_1+1-i_1} \leftarrow R1_{i_1} = (i_1 = n_1,\ldots,1)\ ;\ k = k+n_1$$

$$t^{24}\colon\ W_{k+n_2+1-i_2} \leftarrow R2_{i_2}\ (i_2 = n_2,\ldots,1)\ :\ k = k+n_2\ \}$$

$$Q = \{q^{20}, q^{21}, q^{22}, q^{f}\}$$

and the mapping δ

$$\delta\ (q^{20},) = (< \text{instruction} > q^{21}, t^{11}) \tag{55}$$

$$\delta\ (q^{21}, \{RO_1 = RO_2 = \emptyset\}) = (q^{f}) \tag{56}$$

$$\delta(q^{21}, \{(RO_1 = 0 \wedge R1_1 \equiv \underline{P}) \vee (RO_1 = 1 \wedge RO_2 = 1 \wedge R1_1 \equiv \overline{P})\}) = (q^{22}, t^{21}) \tag{57}$$

$$\delta\ (q^{21},\ \{(RO_1 = 1 \wedge R1_1 \equiv \underline{P}) \vee (RO_1 = 0 \wedge R1_1 \equiv \overline{P})\}) = (q^{22}) \tag{58}$$

$$\delta\ (q^{21},\ \{RO_1 = 1 \wedge RO_2 = 0 \wedge R1_1 \equiv \overline{P}\}) = (q^{22}, t^{23}) \tag{59}$$

$$\delta\ (q^{22},\ \{(RO_2 = 0 \wedge R2_1 \equiv \underline{P}) \vee (RO_2 = 1 \wedge RO_1 = 1 \wedge R2_1 \equiv \overline{P})\}) = (q^{20}, t^{22}) \tag{60}$$

$$\delta\ (q^{22},\ \{(RO_2 = 1 \wedge R2_1 \equiv \underline{P}) \vee (RO_2 = 0 \wedge R2_1 \equiv \overline{P})\}) = (q^{20}) \tag{61}$$

$$\delta\ (q^{22},\ \{RO_2 = 1 \wedge RO_1 = 0 \wedge R2_1 \equiv \overline{P}\}) = (q^{20}, t^{24}) \tag{62}$$

The transfers of M^{b4} simply move the content of R1 or R2 into W, as they are or in reverse order. At q^{20}, Mb^4 as M^{b3}, accepts instructions with respect to which curve set is desired. Each instruction is coded and stored in RO_1 and RO_2 as before. Then depending on the content of RO_1, RO_2 and the colors of the closing points, M^{b4} proceeds according to the table in Figure 2.4. When no further instructions are given and $RO_1 = RO_2 = \emptyset$, M^{b4} is done and goes to its final state q^f.

9. Discontinuous Domains

Our discussion to this point referred to domains delineated by single curves. In this section we outline how the algorithm may be applied to discontinuous domains, that is, domains delineated by more than one curve. The presentation is informal.

The Bead Machine operates upon two curves at a time. It is applicable to pairs of discontinuous domains if their curves are properly paired. Then M^b goes through multiple cycles, each time operating on a single pair. In general each and all curves of the one domain should be paired and operated upon with each and all curves of the other domain. The pairing sequence should be in accordance with the hierarchical structure of the domains.

A domain's hierarchical structure can be represented by properly relating the curves of the domain with union and/or intersection operators. For example, the domains D^1 (Figure 9.1 (a)) and D^3 (Figure 9.2 (a)) can be written as

$$D^1 = (C^{11} \cap C^{12}) \cup (C^{13} \cap C^{14}) \quad \text{and} \quad D^3 = (C^{31} \cap (C^{32} \cup C^{33})) \cup C^{34} \cup C^{c35}.$$

The above are referred as relational expressions and are used to denote the hierarchical order of domains.

<u>Figure 9.1 about here</u> <u>Figure 9.2 about here</u>

Notice that the relational expressions of domains D^2 (Figure 9.1 (b)) and D^4 (Figure 9.2 (b)) are $D^2 = C^{21}$ and $D^4 = C^{45}$. Since the inner space delineated by the curves C^{22}, C^{23} and C^{24} are subjoint to the inner space delineated by C^{21}, the latter suffices for the representation of D^2. Similarly, C^{45} suffices for the representation of D^4 since C^{41}, C^{42}, C^{43} and C^{44} are redundant. In such cases we say that a domain's curve set is <u>inconsistent</u>. When there is no redundancy among the curves delineating a domain, the curve set is <u>consistent</u>; for example, D^1 and D^2 above. A domain's curve set should be consistent; if not it should be simplified as we did for D^2 and D^4 above. Procedures can be defined which, by distinguishing and depicting the levels at which a domain's curves lie, can check their consistency and derive their relational expressions. They will not be discussed in this paper.

To derive the union, intersection or difference of two discontinuous domains, we first derive the order of their curve sets and their relational expressions. In doing this we also check for their consistency. We then join their expressions with the appropriate operator ($\cup$, $\cap$ or -) and apply a sequence of transformations, permissible by the set theoretic laws. Our task is to properly mix the curves of the domains and separate the unions from the intersections. If we are deriving the union of two discontinuous domains, then we are looking for an expression of the form $(C^{ai} \cup C^{bj}) \cap (C^{ai'} \cup C^{bj'})$ where C^{ai} and C^{bj} form different domains. If the intersection, for an expression of the form $(C^{ai} \cap C^{bj}) \cup (C^{ai'} \cap C^{bj'})$. For the differences we apply the set theoretic definition of the difference $(A-B = A\cap\overline{B})$ and execute them as intersections.

<u>Figure 9.3 about here</u>

For example, assume we are given the domains shown in Figure 9.3. That is

$$D^a = \{C^{a1}, C^{a2}\} = (C^{a1} \cap C^{a2}) \quad \text{and} \quad D^b = \{C^{b1}, C^{b2}, C^{b3}\} = ((C^{b1} \cap C^{b2}) \cup C^{b3})$$

where

$$C^{a1} = (\underline{P}_0^{a1}, P_1^{a1}, P_2^{a1}, P_3^{a1}, \underline{P}_0^{a1}) \quad \text{and} \quad C^{a2} = (\underline{P}_0^{a2}, P_1^{a2}, P_2^{a2}, P_0^{a2})$$

$$C^{b1} = (\underline{P}_0^{b1}, P_1^{b1}, P_2^{b1}, P_3^{b1}, P_4^{b1}, \underline{P}_0^{b1}), \quad C^{b2} = (\underline{P}_0^{b2}, P_1^{b2}, P_2^{b2}, \underline{P}_0^{b2})$$

$$C^{b3} = (\underline{P}_0^{b3}, P_{11}^{b3}, P_2^{b3}, \underline{P}_0^{b3}).$$

Then $D^a \cup D^b = (C^{a1} \cap C^{a2}) \cup ((C^{b1} \cap C^{b2}) \cup C^{b3}) =$

$$(C^{a1} \cup C^{b1} \cup C^{b3}) \cap (C^{a2} \cup C^{b1} \cup C^{b3}) \cap (C^{a1} \cup C^{b2} \cup C^{b3}) \cap (C^{a2} \cup C^{b2} \cup C^{b3}),$$

$$D^a \cap D^b = (C^{a1} \cap C^{a2}) \cap ((C^{b1} \cap C^{b2}) \cup C^{b3}) =$$

$$= (C^{a1} \cap C^{a2} \cap C^{b1} \cap C^{b2}) \cup (C^{a1} \cap C^{a2} \cap C^{b3}),$$

$$D^a - D^b = D^a \cap \bar{D}^b = (C^{a1} \cap C^{a2}) \cap ((\bar{C}^{b1} \cup \bar{C}^{b2}) \cap \bar{C}^{b3} =$$

$$= (C^{a1} \cap C^{a2} \cap \bar{C}^{b1} \cap \bar{C}^{b3}) \cup (C^{a1} \cap C^{a2} \cap \bar{C}^{b2} \cap \bar{C}^{b3}),$$

$$D^b - D^a = D^b - \bar{D}^a = ((C^{b1} \cap C^{b2}) \cup C^{b3}) \cap (\bar{C}^{a1} \cup \bar{C}^{a2}) =$$

$$(\bar{C}^{a1} \cap C^{b1} \cap C^{b2}) \cup (\bar{C}^{a2} \cup C^{b1} \cap C^{b2}) \cup (\bar{C}^{a1} \cap C^{b3}) \cup$$

$$(\bar{C}^{a2} \cap C^{b3}).$$

The above expressions can be readily executed on the Bead Machine to derive the curve sets they represent. We execute the operations within the parentheses from left to right. Each pair of parentheses derives a curve of the curve set.

In defining the difference operations, we have relied on the fact that $A-B = A \cap \bar{B}$ where $\bar{B}$ is the complement of B. Our notation for the ordering of curves is the same, that is, the negatively ordered curve $\bar{C}$ is the complement of C (positively ordered). This provides the basis for an alternative derivation of the difference operations from that found in section 6.

10. CONCLUSION

Figure (10.1) illustrates how the same pair of curves can be equally well approximated with a regular (10.1 (b)) and an irregular (10.1 (c)) pair of point vectors.

Figure 10.1 about here

The derivation of the union, intersection and the differences is much simpler for a regular conjunction than for irregular ones. Figure 10.1 suggests that an irregular pair might easily be transformed into a regular, by slightly incrementing the x and y values of the c-points which cause the pair's irregularity. Since we expect the algorithm to be applied in problem areas involving multiple operations, the small transformation could easily accumulate a non-negligible error. Thus, we have chosen to develop it general enough to resolve regular as well as irregular conjunctions.

The algorithm is efficient. Only the expansions of the curves require computation. The characterization of points and their threading require only transfers and testing. In the analysis of efficiency, transfers of single variables can be ignored; only transfers involving all points of one of the curves needs to be considered, e.g., transfers t^1 and t^2.

More specifically, the operations required for normalized curves are:

	Operations:	Transfers:
M^{b1}	$(n_1-1+\beta)(n_2-1+\beta)$	$(n_1-1+\beta)(n_2-1+\beta)$
M^{b2}		$(n_1-1+\beta)+(n_2-1+\beta)$
M^{b3}		$\alpha(n_1+n_2+2b)$

where α denotes the number of disjoint curves derived from the expansions and β denotes the number of S points added during the expansion.

Operations increase geometrically with n_1 and n_2.

APPENDIX I:

A. Function INS

In general, two non-parallel lines, each defined by a pair of points, intersect at

$$x^* = \frac{(x_1^2-x_2^2)(x_1^1y_2^1-x_2^1y_1^1) - (x_1^1-x_2^1)(x_1^2y_2^2-x_2^2y_1^2)}{(y_1^1-y_2^1)(x_2^2-x_1^2) - (y_1^2-y_2^2)(x_2^1-x_1^1)}$$

$$y^* = \frac{(y_1^1-y_2^1)(x_1^2y_2^2-x_2^2y_1^2) - (y_1^2-y_2^2)(x_1^1y_2^1-x_2^1y_1^1)}{(x_1^2-x_2^2)(y_1^1-y_2^1) - (x_1^1-x_2^1)(y_1^2-y_2^2)}$$

These functions are used throughout and shall be called INO. For our purposes

$P^* \leftarrow$ INO $(P_1^1,P_2^1,P_1^2,P_2^2)$ where $P^* = (x^*,y^*)$ or $P^* = \emptyset$ (if the lines are parallel).

Function INS is as follows:

INS $(P_1^1,P_2^1,P_1^2,P_2^2) = P^* = (x^*,y^*)$ if $P^* \neq \emptyset$ and

$(x_1^1 > x^* > x_2^1$ or $x_1^1 < x^* < x_2^1)$ and $(y_1^1 > y^* > y_2^1$ or $y_1^1 < y^* < y_2^1)$ and

$(x_1^2 > x^* > x_2^2$ or $x_1^2 < x^* < x_2^2)$ and $(y_1^2 > y^* > y_2^2$ or $y_1^2 < y^* < y_2^2)$;

$P^* = \emptyset$ otherwise.

B. Function INT

For the definition of INT we shall use function INO again, and also the functions KPR and ORS which will be defined first.

(i) KPR $(P_1^1,P_2^1,P_3^1,P_1^2,P_2^2,P_3^2) = (K,T_1,T_2)$ where KPR checks what kind of a pair (P_1^1,P_2^1) and (P_1^2,P_2^2) are and

$K\leftarrow 1$, $T_1 = T_2 \leftarrow 0$ if it is a non-intersecting pair;

$K\leftarrow 2$, $T_1 = T_2 \leftarrow 0$ if it is an intersecting pair;

$K\leftarrow 3$, if it is a tangent pair and

$T_1\leftarrow 1$, $T_2\leftarrow 0$ if one sided tangency and P_2^1 the tangent point,

$T_1\leftarrow 0$, $T_2\leftarrow 1$ if one sided tangency and P_2^2 the tangent point, and

$T_1\leftarrow 1$, $T_2\leftarrow 1$ if two sided tangency and P_2^1 and P_2^2 the tangent points.

KPR's computations are as follows:

$P^*\leftarrow$ INO $(P_1^1,P_2^1,P_1^2,P_2^2)$.

Where if $P^* = \emptyset$, then $K\leftarrow 1$ and $T_1 = T_2\leftarrow 0$

if $P^* = (x^*,y^*)$ and $P^* = P_2^1$ and $P^* \neq P_2^2$, then $K\leftarrow 3$, $T_1\leftarrow 1$, $T_2\leftarrow 0$;

if $P^* = (x^*,y^*)$ and $P^* = P_2^2$ and $P^* \neq P_2^1$, then $K\leftarrow 3$, $T_1=0$, $T_2\leftarrow 1$;

if $P^* = (x^*,y^*)$ and $P^* = P_2^1 = P_2^2$ then $K\leftarrow 3$, $T_1\leftarrow 1$, $T_2\leftarrow 1$;

if $(x_1^1 > x^* > x_2^1$ or $x_1^1 < x^* < x_2^1)$ and $(y_1^1 > y^* > y_2^1$ or $y_1^1 < y^* < y_2^1$

and $(x_1^2 > x^* > x_2^2$ or $x_1^2 < x^* < x_2^2)$ and $(y_1^2 > y^* > y_2^2$ or $y_1^2 < y^* < y_2^2)$

and $P^* \neq P_2^1$, $P^* \neq P_2^2$, then $K\leftarrow 2, T_1\leftarrow T_2\leftarrow 0$;

else $K = 1$, $T_1 = T_2 = 0$.

(ii) ORS $(P_0,P_1,P_2,P_3,P_4) = (c_1,c_2)$ where ORS orders the segments $L^1 = (P_0,P_1)$, $L^2 = (P_0,P_2)$, $L^3 = (P_0,P_3)$ and $L^4 = (P_0,P_4)$ and then depicts the colors (c_1,c_2) of P_0. c_1 is the color of P_0 in (P_1,P_0,P_2) and c_2 the color of P_0 in (P_3,P_0,P_4). Each c can be

S or $\underline{P}$ or $\overline{P}$ or P.

To order the segments, ORS first figures out the cycle quadrant at which each segment lies. The center of the cycle is at P_0 and g, the index of its quadrants is as in Figure I.1. Then

$$g^j \leftarrow 1 \text{ if } x_o \leq x_j \wedge y_o < y_j; \quad g^j \leftarrow 2 \text{ if } x_o > x_j \wedge y_o \leq y_j;$$

$$g^j \leftarrow 3 \text{ if } x_o \geq x_j \wedge y_o > y_j; \quad g^j \leftarrow 4 \text{ if } x_o < x_j \wedge y_o \geq y_j.$$

The segments are ordered according to the numeric values of their g's, where 4 is followed by 1. For equal g's, the segments are ordered according to their slopes. That is if, say, $g^1 = g^3$, then

$$s_1 \leftarrow (y_1 - y_o) \,/\, (x_1 - x_o); \; s_3 \leftarrow (y_3 - y_o) \,/\, (x_3 - x_o)$$

and $L^1 \prec L^3$ if $(s_1 < s_3$ and $g^1 = g^3 = 1$ or 3) or $(s_1 > s_3$ and $g^1 = g^3 = 2$ or 4);

$L^1 \succ L^3$ if $(s_1 > s_3$ and $g^1 = g^3 = 1$ or 3) or $(s_1 < s_3$ and $g^1 = g^3 = 2$ or 4);

$L^1 \approx L^3$ if $s_1 = s_3$ and $g^1 = g^3 = 1$, 2, 3 or 4.

The signs $\prec, \succ$, and $\approx$ mean proceeds, follows and coincides respectively, and $L^1 \succ L^3 = L^3 \prec L^1$. For example, the segments in Figure I.2 will be ordered as $L^2 \prec L^3 \prec L^1 \prec L^4$ since $g^2=1$, $g^3=g^1=3$, $g^4=4$ and $s_3 < s_1$.

<u>Figure I.1 about here</u> <u>Figure I.2 about here</u>

We shall simplify the notation $L^2 \prec L^3 \prec L^1 \prec L^4$ by eliminating the L's and writing their subscript indicators only. We shall also request that L^1 (now simply 1) is listed first and also last; the latter to indicate the ordering's circularity. Then $L^2 < L^3 < L^1 < L^4$ will be written as $1 < 4 < 2 < 3 < 1$. Then

$c_1 = c_2 \leftarrow S$ if $1 < 3 < 2 < 4 < 1$ or $1 < 4 < 2 < 3 < 1$;

$c_1 \leftarrow \underline{P}$ and $c_2 \leftarrow \overline{P}$ if $1 < 3 < 4 < 2 < 1$ or $1 < 3 < 4 \approx 2 < 1$ or $1 \approx 3 < 4 < 2 < 1$;

$c_1 \leftarrow \overline{P}$ and $c_2 \leftarrow \underline{P}$ if $1 < 2 < 4 < 3 < 1$ or $1 < 2 < 4 < 3 \approx 1$ or $1 < 2 \approx 4 < 3 < 1$;

$c_1 = c_2 \leftarrow \underline{P}$ if $1 < 2 < 3 < 4 < 1$ or $1 < 2 < 3 < 4 \approx 1$ or $1 < 2 \approx 3 < 4 < 1$

$c_1 = c_2 \leftarrow \overline{P}$ if $1 < 4 < 3 < 2 < 1$ or $1 < 4 < 3 \approx 2 < 1$ or $1 \approx 4 < 3 < 2 < 1$

$c_1 = c_2 \leftarrow P$ if $1 \approx 3 < 2 \approx 4 < 1$ or $1 \approx 4 < 2 \approx 3 < 1$.

As before, S is the color for an intersection point, $\overline{P}$ the color of an inner P and $\underline{P}$ the color for an outer P.

We can now define INC as follows

$$\text{INT } (P_1^1, P_2^1, P_3^1, P_1^2, P_2^2, P_3^2) = (P,\ c_1,\ c_2) \text{ where}$$

$P \leftarrow \emptyset$, $c_1 = c_2 \leftarrow \emptyset$ if KPR $(P_1^1, P_2^1, P_3^1, P_1^2, P_2^2, P_3^2) = (1,0,0)$

$P \leftarrow P^* = (x^*, y^*)$, $c_1 = c_2 \leftarrow S$ if KPR $(P_1^1, P_2^1, P_3^1, P_1^2, P_2^2, P_3^2) = (2,0,0)$

$P \leftarrow P_2^1$, $(c_1, c_2) \leftarrow$ ORS $(P_2^1, P_1^1, P_3^1, P_1^2, P_3^2)$ if KPR $(P_1^1, P_2^1, P_3^1, P_1^2, P_2^2, P_3^2) = (3,1,0)$

$P \leftarrow P_2^2$, $(c_1, c_2) \leftarrow$ ORS $(P_2^2, P_1^1, P_2^1, P_1^2, P_3^2)$ if KPR $(P_1^1, P_2^1, P_3^1, P_1^2, P_2^2, P_3^2) = (3,0,1)$

$P \leftarrow P_2^1$, $(c_1, c_2) \leftarrow$ ORS $(P_2^1, P_1^1, P_3^1, P_1^2\ P_3^2)$ if KPR $(P_1^1, P_2^1, P_3^1, P_1^2, P_2^2, P_3^2) = (3,1,1)$

Notice that when we outlined INT (Section 4) we used a slightly different notation for the function's outcome. The two notations are equivalent. For example, (P^S, P^S) is equivalent with (P, c_1, c_2) where $P = P_2^1 = P_2^2$ and $c_1 = c_2 = S$.

APPENDIX II:

The Normalizer M^{b0}

The normalizer M^{b0} is the system (L,K,T,G,Q,δ) where L as before,

$K = \{k^{01}$: $R0_2 \leftarrow PON\ (R1_{n_1}, R2_{n_2}, R2_1$

k^{02}: $R0_1 \leftarrow MDP\ (R1_{n_1}, R1_1)$

k^{03}: $R0_1 \leftarrow NIN\ (R0_1, R1_1, R0_2, R2_{n_2}, R2_1, R2_2)$

k^{04}: $R0_2 \leftarrow CPY\ (R1_{n_1})\}$

$T = \{t^1, t^2, t^3, t^9, t^{11}$ as before and

t^{01}: $R0_2 \leftarrow R2_1;\ R2_{i_2} \leftarrow R2_{i_2+1}\ (i_2 = 1, \ldots, n_2 - 1);\ n_2 = n_2 - 1$

t^{02}: $R1_{i_1+1} \leftarrow R1_{i_1}\ (i_1 = 1, \ldots, n_1 (\mathrm{mod}\ n_1))\}$

$G = \{c^1, c^3$ as before$\}$

$Q = \{q^{00}, q^{01}, q^{02}, q^{03}, q^{04}, q^{05}, q^{06}, q^{07}, q^{0f}, q^{0ff}\}$

and δ the following mapping:

$$\delta\ (q^{00}, \{\ \}) = (q^{01}, t^1, t^2, k^{01}) \tag{1}$$

$$\delta\ (q^{01}, \{R0_2 = 1\}) = (q^{02}, t^1) \tag{2}$$

$$\delta\ (q^{01}, \{R0_2 = 0 \wedge R2_{n_2} \neq R2_1\}) = (q^{01}, t^2, k^{01}) \tag{3}$$

$$\delta\ (q^{01}, \{R0_2 = 0 \wedge R2_{n_2} = R2_1\}) = (q^{05}, k^{04}, t^3, t^1) \tag{4}$$

$$\delta\ (q^{02},\ \{R1_{n_1} = R1_1\}) = (q^{03}, t^1, k^{02}, t^3) \tag{5}$$

$$\delta\ (q^{02},\ \{R2_{n_2} = R2_1\}) = (q^{01}, t^2, k^{01}) \tag{6}$$

$$\delta\ (q^{02},\ \{R2_{n_2} \neq R2_1\}) = q^{02}, t^2) \tag{7}$$

$$\delta\ (q^{03},\ \{R1_{n_1} = R1_1\}) = (q^{0ff}) \tag{8}$$

$$\delta\ (q^{03},\ \{R2_{n_2} = R2_1\}) = (q^{04}, t^2, k^{01}) \tag{9}$$

$$\delta\ (q^{03},\ \{R2_{n_2} = R2_1\}) = (q^{03}, t^2) \tag{10}$$

$$\delta\ (q^{04},\ \{RO_2 = 1\}) = (q^{03}, t^9, t^1, k^{02}, t^3) \tag{11}$$

$$\delta\ (q^{04},\ \{RO_2 = 0 \wedge R2_{n_2} \neq R2_1\}) = (q^{04}, t^2, k^{01}) \tag{12}$$

$$\delta\ (q^{04},\ \{RO_2 = 0 \wedge R2_{n_2} = R2_1\}) = (q^{05}, k^{04}, t^3, t^1) \tag{13}$$

$$\delta\ (q^{05},\ \{R1_{n_1} \neq R1_1\}) = (q^{05}, t^1) \tag{14}$$

$$\delta\ (q^{05},\ \{R1_{n_1} = R1_1\}) = (q^{06}, t^9) \tag{15}$$

$$\delta\ (q^{06},\ \{R1_{n_1} \neq R1_1\}) = (q^{06}, t^1) \tag{16}$$

$$\delta\ (q^{06},\ \{R1_{n_1} = R1_1\}) = (q^{07}, t^{01}, k^{03}) \tag{17}$$

$$\delta\ (q^{07},\ \{R2_1 \neq RO_2\}) = (q^{07}, t^2, k^{03}) \tag{18}$$

$$\delta\ (q^{07},\ \{R2_1 = RO_2 \wedge RO_1 = 0\}) = (q^{0f}, c^3, t^{02}, c^3, t^1) \tag{19}$$

$$\delta\ (q^{07},\ \{R2_1 = RO_2 \wedge RO_1 = 1\}) = (q^{0f}, c^1, t^{02}, c^1, t^1) \tag{20}$$

The function PON (P^0, P^1_1, P^1_2) checks if the point P^0 lies on the segment (P^1_1, P^1_2). Its value is 1 for yes, and 0 for no. More precisely

$$\text{PON } (P^0, P_1^1, P_2^1) = 1 \quad \text{if} \quad y^0 = (A^1x^0 + C^1)/(-B^1) \text{ and } (x_1^1 \leq x^0 \leq x_2^1 \text{ or } x_1^1 \geq x^0 \geq x_2^1)$$
$$\text{and } (y_1^1 \leq y^0 \leq y_2^1 \text{ or } y_1^1 \geq y^0 \geq y_2^1)$$
$$= 0 \text{ otherwise.}$$

$A^1x-B^1y-C^1=0$ is the equation of (P_1^1, P_2^1) and $(x^0, y^0) = P^0$, $(x_1^1, y_1^1) = P_1^1$ and $(x_2^1, y_2^1) = P_2^1$.

The function MDP (P_1, P_2) calculates the midpoint of the segment (P_1, P_2). That is

$$\text{MDP } (P_1, P_2) = P_m = (x_m, y_m) \text{ where } x_m = (x_1 + x_2)/2 \text{ and } y_m = (y_1 + y_2)/2.$$

The function NIN $(R, P_1^1, P_2^1, P_1^2, P_2^2, P_3^2)$ checks if the segment (P_1^2, P_2^2) intersects the line defined by (P_1^1, P_2^1) at a point before P_1^1. If yes and R=1, or no and R=0, the output is 0. If no and R=1, or yes and R=0, the output is 1. P_3^2 is needed in case P_2^2 is a tangent point.

The function CPY (P) simply copies P. That is CPY (P) = P.

M^{b0} normalizes the point vector in R1. Its first part (map. 2-4) checks if the given closing point is unique (if it is not a point of the other curve also). If it is not, it picks the next point in the R1 vector (map. 2), resets R2 (map. 6,7) and proceeds to check if this point is unique. It repeats the process till it depicts some unique point in R1. If R1 completes a cycle and no unique point is found (map. 5), M^{b0} goes to state q^{03} and starts creating points by taking the midpoints of the segments in R1. It creates one point per segment and applies tests as before (map. 12, 13), till some of the created points are found unique. M^{b0} will not find a unique point if the curves are coincident. In such a case it goes to q^{0ff}, the final state for failure and exits.

M^{b0} will also fail to depict a unique point in few extreme cases like the one shown in Figure II.1. Surely its definition can be made stricter by pikcing more points on each segment. It was thought unnecessary since such cases are very unlikely to occur.

Figure II.1 about here

Whenever a unique point is depicted, M^{b0} goes to state q^{05} and then q^{06} where it resets the vectors in R1 and R2, so that their closing points occupy locations $R1_{n_1}$, $R1_1$, $R2_{n_2}$ and $R2_1$ (map. 14-17) and goes to state q^{07}. At q^{07}, it checks how many segments of R2 are intersected by the line defined by the closing points in R1 and R2 (now stored in RO_2) at a point before the closing point of R1. If it is an odd number (when RO_1=1), the closing point in R1 is an inner point; it is outer if the number of intersections is even (when RO_1=0) (map. 18-20). The final state for success is q^{0f}.

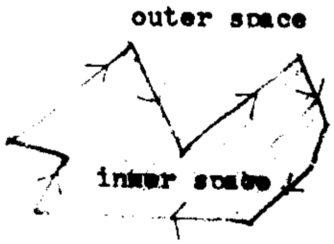


Figure 2.1

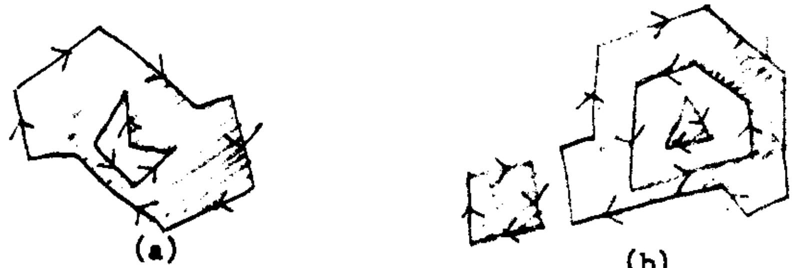


Figure 2.2

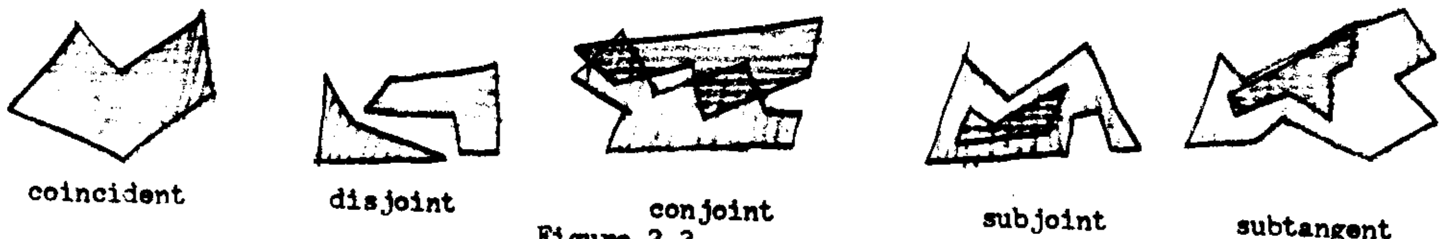


Figure 2.3

	DISJOINT	SUBJOINT C^2 in C^1	SUBJOINT C^1 in C^2	COINCIDENT	CONJOINT
$C^1 \cup C^2$	$\{C^1, C^2\}$	$\{C^1\}$	$\{C^2\}$	$\{C^1\}$ or $\{C^2\}$	$f^{\cup}(C^1, C^2)$
$C^1 \cap C^2$	$\{\emptyset\}$	$\{C^2\}$	$\{C^1\}$	$\{C^1\}$ or $\{C^2\}$	$f^{\cap}(C^1, C^2)$
$C^1 - C^2$	$\{C^1\}$	$\{C^1, \overline{C}^2\}$	$\{\emptyset\}$	$\{\emptyset\}$	$f^{-}(C^1, C^2)$
$C^1 - C^2$	$\{C^2\}$	$\{\emptyset\}$	$\{C^2, \overline{C}^1\}$	$\{\emptyset\}$	$f^{-}(C^2, C^1)$

Figure 2.4

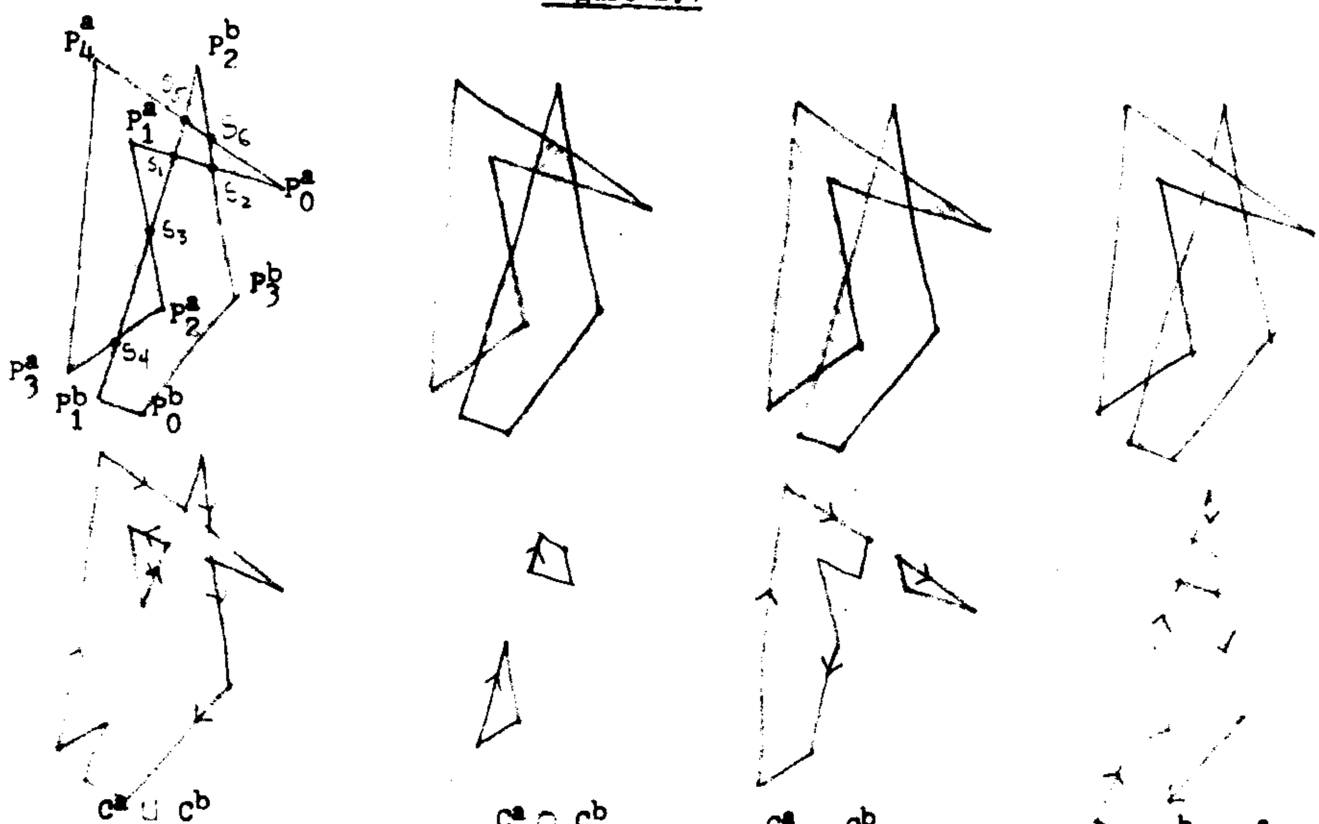


Figure 2.5

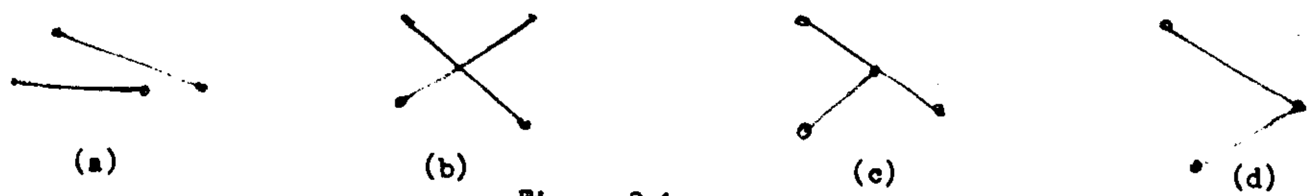


Figure 3.1

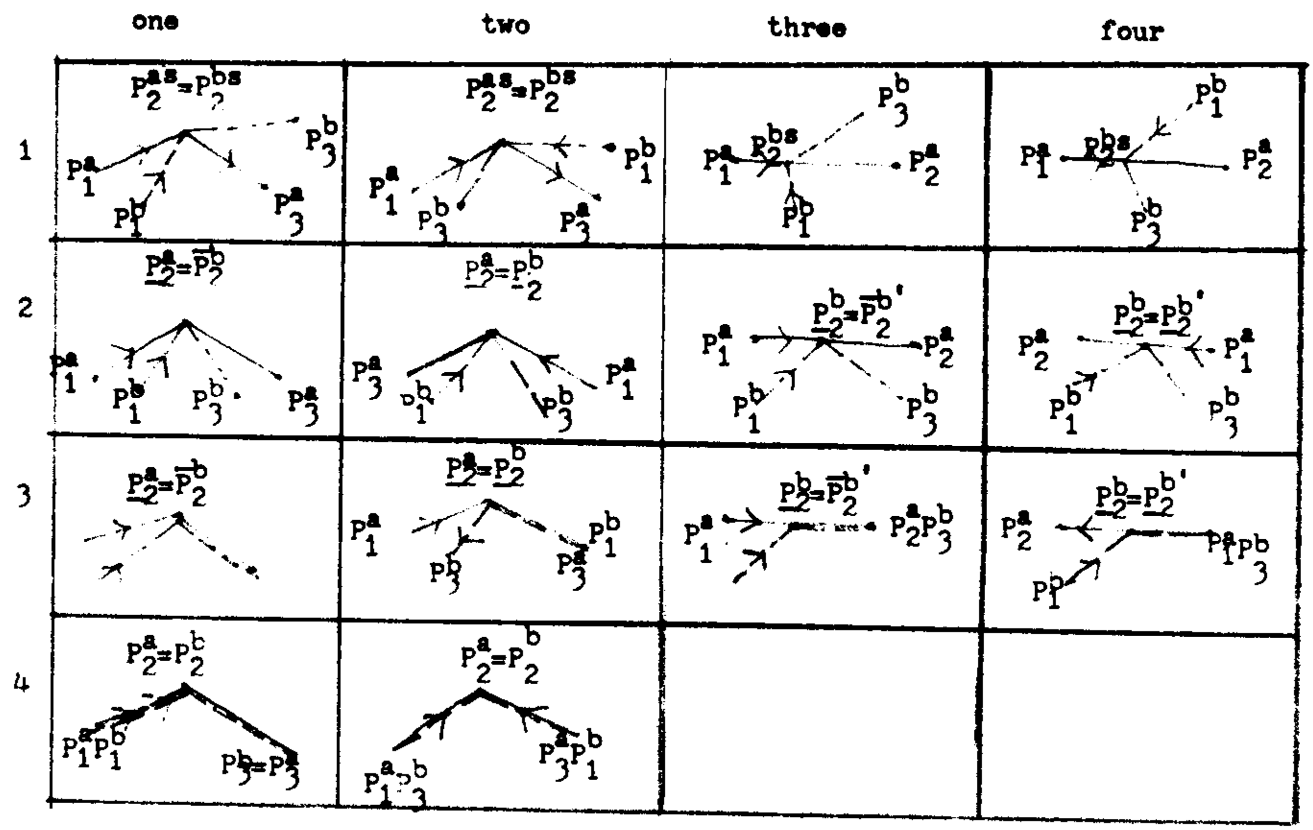


Figure 4.1

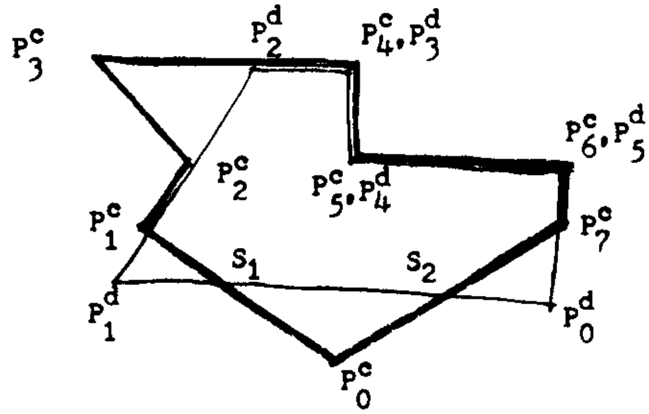


Figure 4.2

	B in $C^{ab'}$	B in $C^{ba'}$
$C^a \cup C^b$	$S\ \underline{P}$ or $\bar{P}\ S$	
$C^a \cap C^b$	$S\ \bar{P}$ or $\underline{P}\ S$	
$C^a - C^b$	$S\ \underline{P}$ or $\bar{P}\ S$	$S\ \bar{P}$ or $\underline{P}\ S$
$C^b - C^a$	$S\ \bar{P}$ or $\underline{P}\ S$	$S\ \underline{P}$ or $\bar{P}\ S$

(a)

$C^{ab'}$	$C^{ba'}$
1	1
0	0
1	0
0	1

(b)

Figure 6.1

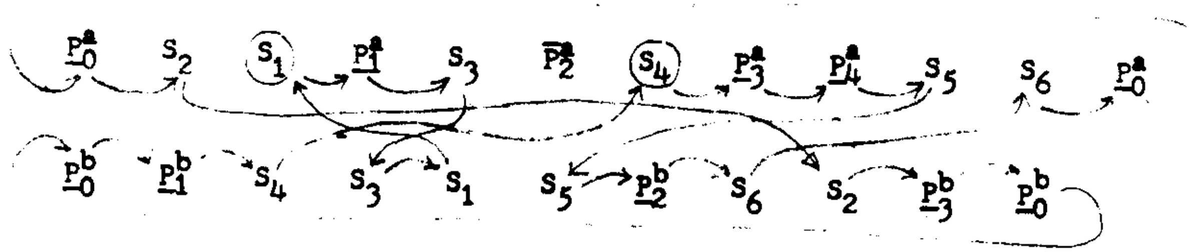


Figure 6.2

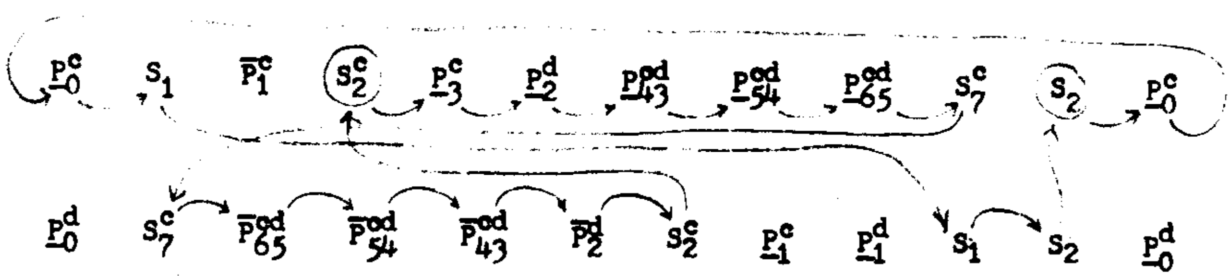


Figure 6.3

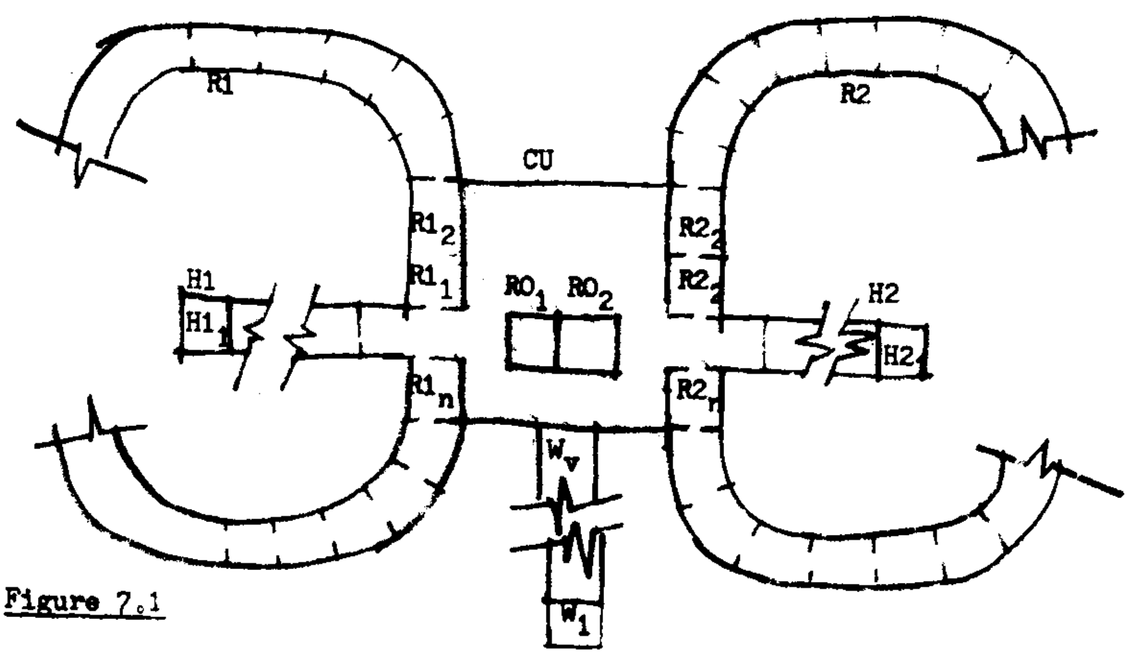
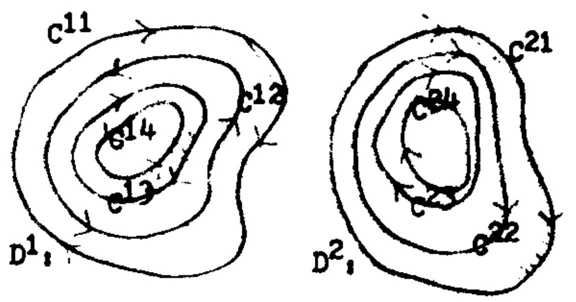
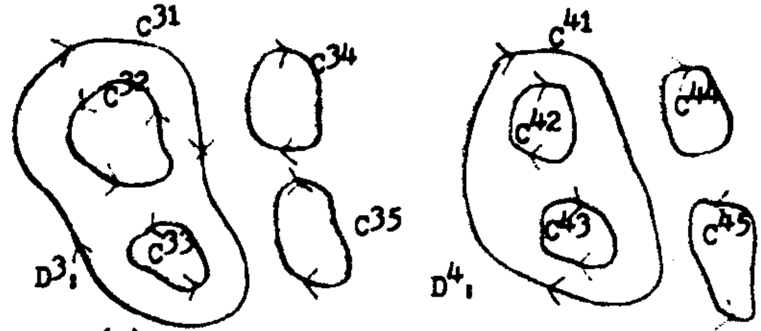


Figure 7.1



(a) (b)

Figure 9.1



(a) (b)

Figure 9.2

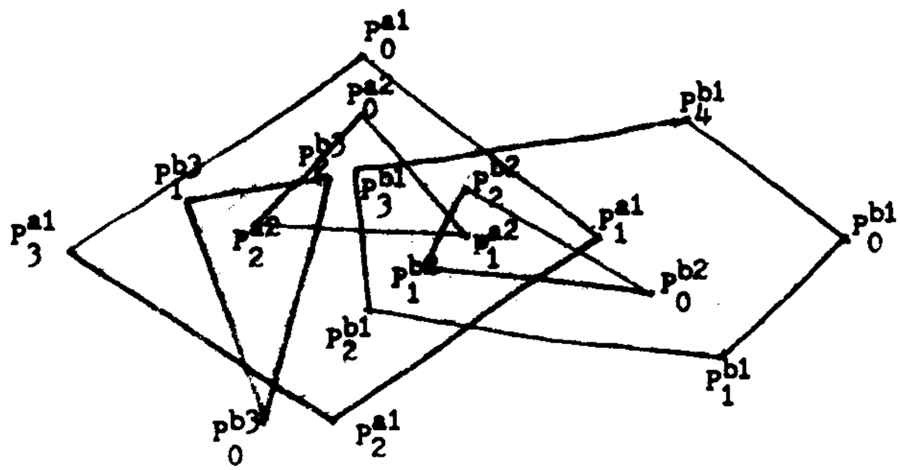
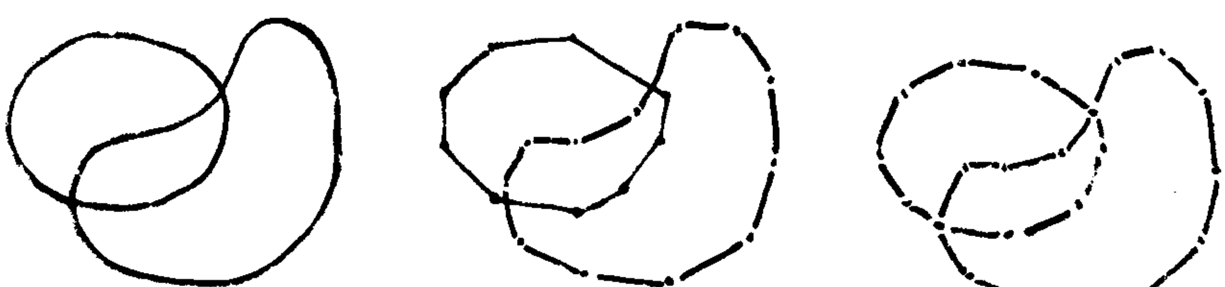


Figure 9.3



(a) (b) (c)

Figure 10.1

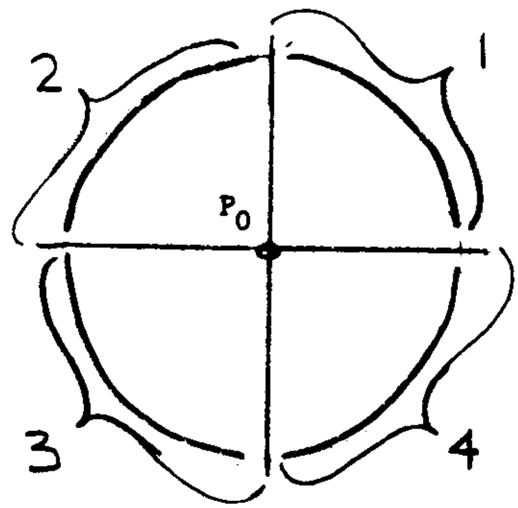


Figure I.1

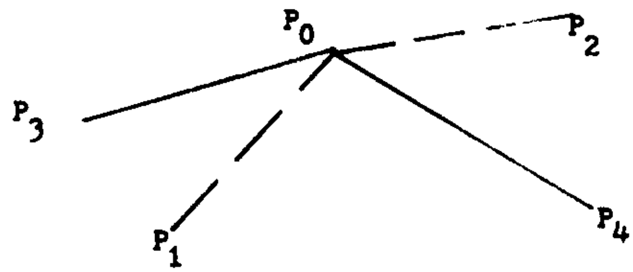


Figure I.2

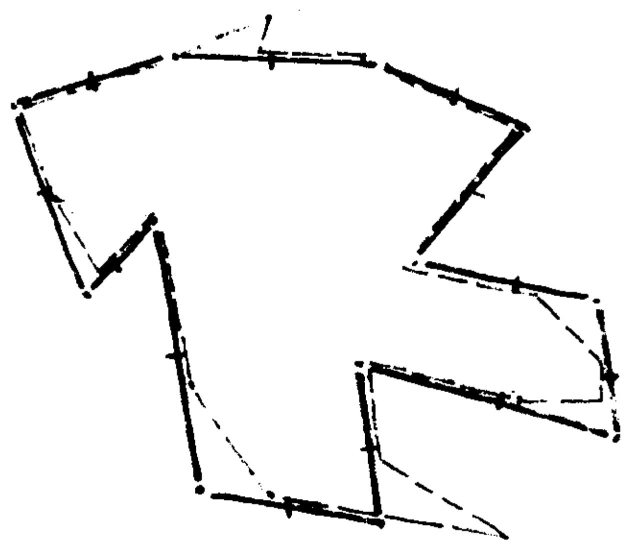


Figure II.1

Bibliography

1. Amidon, Elliot L. MIADS2...an alphanumeric map information assembly and display system for a large computer. U.S. Forest Service Res. Paper PSW-38, 1966.

2. Eastman, C. M. "Heuristic Algorithms for Automated Space Planning," Proceedings of the Second Joint International Conference on Artificial Intelligence, London, 1971.

3. Eastman, C. M. "Representations for Space Planning," Comm. ACM 13,4 (April 1970), 242-250.

4. Horwood, Edgar M., "Computer Applications to Urban Planning and Analysis: Examples and Prospects," (Proceedings IFIPS Conference 1968, North Holland Publishing Co. Amsterdam,) 1040-1052.)

5. Kazmierczak, H., Image Processing and Pattern Recognition, Proceedings IFIPS Conference 1968, North Holland Publishing Co. Amsterdam,)

6. Pfaltz, John L. and A. Rosenfeld, "Computer Representation of Planar Regions by Their Skeletons," Comm. ACM 10,2 (February 1967), 119-123.