INFORMATION AND CONTROL 63, 164–189 (1984)

An Algorithm for Covering Polygons with Rectangles*

D. S. FRANZBLAU AND D. J. KLEITMAN

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Decomposing a polygon into simple shapes is a basic problem in computational geometry, with applications in pattern recognition and integrated circuit manufacture. Here we examine the special case of covering a rectilinear polygon (or polyomino) with the minimum number of rectangles, with overlapping allowed. The problem is *NP*-hard. However, we give here an $O(v^2)$ algorithm for constructing a minimum rectangle cover, when the polygon is vertically convex. (Here v is the number of vertices.) The problem is first reduced to a 1-dimensional interval "basis" problem. In showing our algorithm produces an optimal cover we give a new proof of a minimum basis-maximum independent set duality theorem first proved by E. Györi (*J. Combin Theory Ser. B* 37, No. 1, 1–9).

I. INTRODUCTION

Covering a polygon with the minimum number of rectangles is one of several computationally difficult polygon decomposition problems, surveyed in O'Rourke and Supowit (1983), and Johnson (1982, pp. 189–191). The problem also has practical applications. One is creating a mask for etching an integrated circuit, where polygons represent wires and transistors. (A mask has the same function as a photo negative.) A pattern generator flashes rectangles onto the mask material; the final mask is the union of the rectangles (Chaiken, *et al.*, 1981; Hegedüs, 1982; Mead and Conway, 1980, p. 93). Another application is to storing and displaying figures on a computer terminal (Masek, 1978). Here we shall restrict attention to *rectilinear* or *Manhattan* geometry. We assume polygons are aligned with the x–y coordinate axes, and are polynominoes, finite subsets of unit squares in a grid, with integer vertices.

A *rectangle cover* for a polygon R is a collection of rectangles contained within R, whose union exactly covers R. A *minimum cover* is one with the minimum number of rectangles.

We also assume that rectangles are subsets of squares (aligned with the

* Supported in part by Air Force Contract AFOSR-82-0326A.

0019-9958/84 $3.00

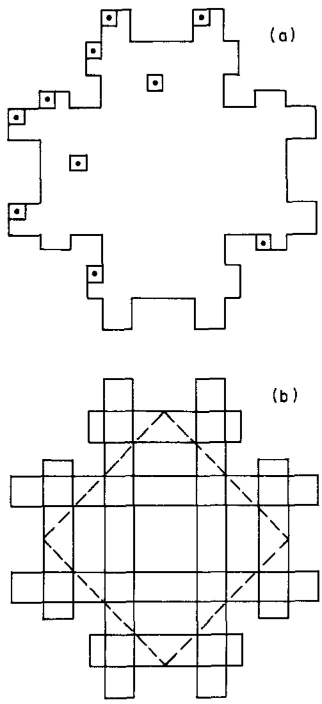


FIG. 1.1. (a) Rectilinear polygon which requires ten vertically aligned rectangles in a minimum cover (since there are ten squares with no two in a common rectangle). (b) Only nine rectangles are required if rotation is allowed.

x–y axes). The example in Fig. 1.1, due to O'Rourke (1982, p. 77), shows that smaller covers can be obtained by allowing rotation of rectangles.

A polygon R is called *simply connected* if it has no holes. R is *vertically convex* if any two squares in the same column of R are joined by a vertical line in R, i.e., every column is connected. Define *horizontally convex* analogously (see Fig. 1.2).

Chvátal originally conjectured that the number of rectangles in a minimum cover of R is equal to the maximum number of squares in R with no two in a common rectangle, as in Fig. 1.1(a). Small counterexamples were contructed by Szemerédi and Chung (in Chaiken *et al.*, 1981), shown in Fig. 1.3(a) and (b) respectively, showing that duality can fail even for simply connected polygons. By contrast, Albertson and O'Keefe (1981) proved a similar duality for covering with *squares*, in the simply connected case.

Chaiken, Kleitman, Saks, and Shearer (1981) proved that duality does hold if the polygon is both vertically and horizontally convex. Recently, Györi (1984) strengthened the result to vertically convex polygons. He

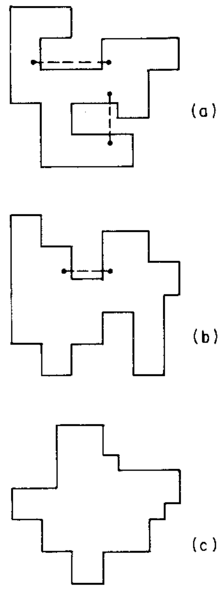


FIG. 1.2. Simply connected rectilinear polygons: (a) neither vertically nor horizontally convex; (b) vertically convex only; (c) convex in both directions.

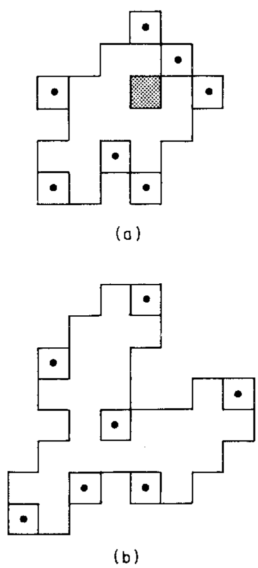


FIG. 1.3. Two polygons each with a minimum cover of size eight but at most seven squares (marked) with no two in a common rectangle: (a) Szemerédi's example (one hole); (b) Chung's simply connected example.

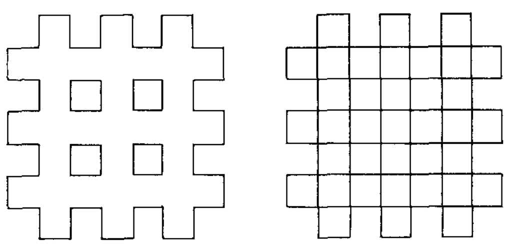
used a reduction to an interval generating set problem, which is also our starting point.

Here we give an efficient, simple algorithm, which constructs a minimum cover for any vertically convex polygon. We also provide a new proof of the duality theorem: the number of intervals in a minimum generating set is equal to the number of intervals in a maximum "independent" subset.

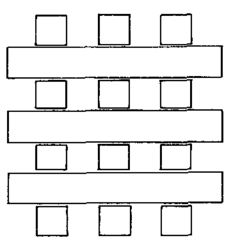
Such minimum basis–maximum independent set duality theorems appear throughout combinatorics, and are often the key to constructing fast optimization algorithms. Aigner (1979) and Lawler (1976) provide good surveys of such results. Covering a polyomino with the minimum number of rectangles has also been studied as a problem in the theory of both graphs and hypergraphs (see Berge *et al.*, 1981; and Saks, 1982).

It is important to distinguish a minimum cover from a minimum *partition* into non-overlapping rectangles. There is a simple $O(v^{5/2})$ algorithm for constructing a minimum partition using bipartite matching, due to Ohtsuki (1982) (where v is the number of vertices of the polygon). (A polynomial-time algorithm has also been found by Pagli *et al.*, 1979, and probably by others.)

However, Masek (1978) has shown that finding a minimum cover for a rectilinear polygon is *NP*-hard (with the corresponding decision problem in *NP*). It is not yet known whether the problem is *NP*-hard when restricted to simply connected polygons.



(a) (b)



(c)

FIG. 1.4. Partitioning versus covering: (a) rectilinear polygon; (b) minimum cover (k rectangles); (c) minimum partition ($k^2/4+k$ rectangles).

Partitioning is a poor heuristic for covering in general. Figure 1.4 shows a polygon (a), a minimum cover (b) and a minimum partition (c). If k is the number of rectangles in the cover, the minimum partition has $k^2/4+k$ rectangles. Franzblau (1985) has shown that the minimum partition is at worst twice the size of the minimum cover plus the number of holes. Thus, partitioning is a reasonable heuristic for simply connected polygons.

For polygons with holes, for all heuristics we have considered, the best known upper bound on the "performance ratio," the ratio of the number of rectangles found by the heuristic to the minimum, k, is $O(\log k)$. In examples, however, the worst ratios are constants (Franzblau, 1985). These heuristics include using the cover obtained from the set of inequivalent horizontal slices (see Section II) and the greedy set-covering algorithm, where the sets are maximal rectangles (see Johnson, 1974; and Chvátal, 1979). An iterated edge-covering heuristic such as in Hegedüs (1982) has not been analyzed for rectilinear polygons. The analysis in Levcopoulos and Lingas (1984) relates the number of rectangles produced by this heuristic, for general (non-rectilinear) polygons, to the number of vertices and the edge lengths. However, their results do not give bounds on the performance ratio.

In the remainder of the paper we describe our algorithm for constructing a minimum rectangle cover for any vertically convex polygon.

In Section II, we show how to reduce the covering problem to finding a minimum generating set (or basis) for a collection of intervals, and define independence for intervals. Section III contains the algorithm for constructing a minimum generating set. The heart of the algorithm is a simple operation on "dependent" sets of intervals, iterated until the set is independent. The important definitions are in the beginning of Sections II and III.

In Section IV we prove that our algorithm is correct, thereby proving the interval version of the duality theorem. The proof depends on several technical lemmas which are proved in the Appendix. We discuss implementation of the algorithm in Section V. The complexity is $O(v^2)$, where v is the number of vertices in the polygon. To assist the reader we list here our most frequently used notation:

R	rectilinear polygon
s	unit square in a grid
r	rectangle (set of squares)
$I, J, K, (X, Y)$	interval (integer endpoints)
x	atom (unit interval $(i, i+1)$)
G, S, T	set (of intervals)
$\|T\|$	cardinality of T
$S-T$	for $T \subseteq S$, $\{I \in S \| I \notin T\}$
$\bigcup S$	union of intervals in S

B, D, (L, R)	bracket (distinguished interval) (Sect. III)
$S(D)$, $S(L, R)$	subset of intervals in S which are subintervals of D or (L, R) (Sect. III)

We use the following convention to distinguish "maximum" and "maximal." If F is a finite family of subsets of a finite set, $f \in F$ is *maximum* if it has maximum cardinality. However, f is *maximal* if no $g \in F$ properly contains f. (If F is partially ordered by inclusion, f is a maximal element of the partial order.) The same distinction applies to minimum and minimal.

II. Reducing Rectangles to Intervals

Let R be a rectilinear polygon. A rectangle $r \subseteq R$ is *maximal* if no other rectangle in R strictly contains r, i.e., all four edges of r contain a boundary segment of R. Call r *vertically maximal* if its upper and lower edges each contain boundary segments of R.

A *horizontal slice* of R is connected set of squares in the same row of R, with both ends on the boundary (i.e., horizontally maximal). Observe that each square in R determines a unique horizontal slice. Each horizontal slice h also determines a unique maximal rectangle in R (the vertically maximal rectangle generated by h). Two horizontal slices which determine the same maximal rectangle will be called *equivalent*. As shown in Fig. 2.1, any *complete* set of inequivalent horizontal slices determines a rectangle cover, since every square is in some slice.

The set of inequivalent horizontal slices can be obtained by sweeping R from top of bottom with a horizontal line, recording intersections with the

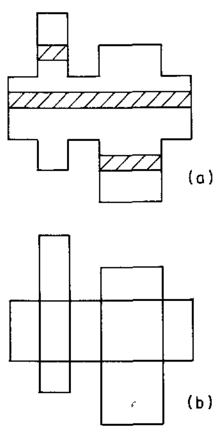


FIG. 2.1. (a) Complete set of inequivalent horizontal slices. (b) Rectangle cover determined by slices.

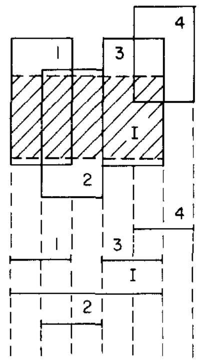


FIG. 2.2. Generating an interval: $I = 1 \cup 2 \cup 3$. Rectangles 1, 2, 3 cover the rectangle determined by I.

vertical boundaries of R, then deleting equivalent slices. Note that slicing yields a simple heuristic for covering any rectilinear polygon (with performance ratio $O(\log k)$ (Franzblau, 1985).

For the remainder of the paper we will restrict attention to vertically convex polygons. A nice observation, due to A. Frank (Györi, 1984), is that the inequivalent horizontal slices provide a compact description of R, as we shall now show. First, project the set of inequivalent slices onto the x-axis, to obtain a set S of real intervals (with integer endpoints). If $I \in S$ and $J \subseteq I$ is any subinterval (not necessarily in S), then J determines a unique vertically maximal rectangle (if there were two, R could not be vertically convex).

In Fig. 2.2 we see that if $J_1, J_2, \ldots, J_k$ are all subintervals of I and $I = J_1 \cup J_2 \cup \cdots \cup J_k$, then all squares in the rectangle determined by I are contained in the union of the rectangles determined by the J_i's.

DEFINITION 2.1. Let S and G be sets of intervals. *G generates S* if every interval in S is equal to the union of one or more intervals in G.

In Fig. 2.2, $\{J_1, J_2, J_3, J_4\}$ generates $\{I, J_1, J_2, J_3, J_4\}$. Observe that any set S generates itself.

We call G a *minimum generating set* if $|G|$ is as small as possible. The following lemma shows how to construct a cover from a generating set and follows directly from the discussion preceding Definition 2.1 and from Fig. 2.2.

LEMMA 2.2. *Let S be the set of intervals determined by R (i.e., by the horizontal slices of R). If G generates S, and G contains only subintervals of intervals in S, then the set of rectangles determined by G covers R.* ▮

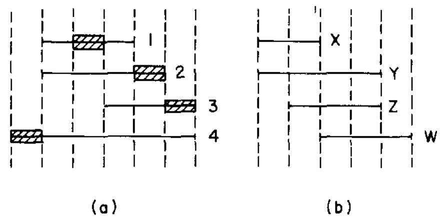


FIG. 2.3. (a) Independent set of intervals, ordered as in Definition 2.4, with corresponding atoms marked. (b) Dependent set: each of x, y, z is contained in the union of the remaining intervals.

COROLLARY 2.3. *The number of rectangles in a minimum cover of R is at most the number of intervals in a minimum generating set for S.* ∎

To complete the translation into an interval problem, we use duality. A subset of squares in R is called an *antirectangle* (or an independent set) if no two squares lie in a common rectangle inside R.

The following interval characterization of independent sets is also due to Frank. Since we shall only be concerned with intervals with integer endpoints, we can think of intervals as finite sets of *atoms*, or intervals of the form $(i, i+1)$.

DEFINITION 2.4. A set of intervals S is *independent* if the intervals can be ordered $I_1, I_2, \ldots, I_n$ so that each I_k contains at least one atom not in $\bigcup_{j=1}^{k-1} I_j$.

A set with no such ordering will be called *dependent.* (A more useful definition is given in the next section.) Figure 2.3 shows an independent and dependent set. Note that if the intervals in S all have distinct left endpoints (or distinct right endpoints) then the set is independent. The converse is false, as Fig. 2.3 shows.

We now show how to construct antirectangles of squares from independent sets of intervals (see Fig. 2.4). Let S be determined by R, and let $T \subseteq S$ be an independent subset, with ordering $(I_1, I_2, \ldots, I_k)$ as in Definition 2.4. For each j, let h_j be any horizontal slice in R projecting onto I_j. Let x_j be an atom on I_j not contained in $I_1\ I_2, \ldots, I_{j-1}$. Finally, let s_j be the unique square in h_j which lies above x_j.

LEMMA 2.5. *The squares $s_1, s_2, \ldots, s_k$ defined above form an antirectangle, i.e., no two lie in a common rectangle.*

Proof (by contradiction). Suppose s_i and s_j are in the same rectangle, with $i<j$. Then the corresponding atoms x_i and x_j must be contained in *both* I_i and I_j. But, by assumption, $x_j \notin I_i$. ∎

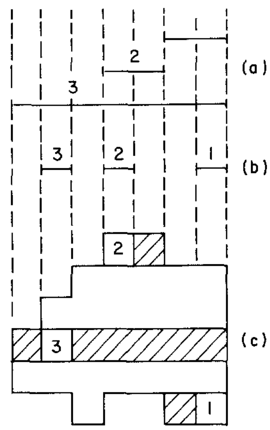


FIG. 2.4. Construction for Lemma 2.5: (a) independent subset of intervals; (b) atoms of Definition 2.4; (c) corresponding horizontal slices (shaded) and squares determined by atoms.

COROLLARY 2.6. *The number of squares in a maximum antirectangle of R is at least the number of intervals in a maximum independent subset of S.*

Let us summarize the results of this section. We now have a top-level description of an algorithm for constructing a minimum rectangle cover:

COVERING ALGORITHM

(1) Find S, the set of intervals determined by inequivalent horizontal slices of R.

(2) Construct a minimum generating set G for S, where each interval in G is a subinterval of some $I \in S$.

(3) Construct the set of vertically maximal rectangles in R determined by intervals in G.

We used the vertical convexity of R in assuming that S is a set (rather than multiset) and in assuming that step (3) is uniquely determined.

Let α represent the cardinality of a maximum independent subset of S, and θ the number of intervals in a minimum generating set.

In addition, let a be the number of squares in a maximum antirectangle of R, and k the number of rectangles in a minimum cover.

It is straightforward to see that $a \leqslant k$. Thus, by Corollaries 2.3 and 2.6 we have

$$\alpha \leqslant a \leqslant k \leqslant \theta.$$

To prove that the covering algorithm produces an optimal cover, we shall show how to construct a generating set G with $|G| = \alpha$, so that $|G| = \theta$.

III. Constructing a Minimum Generating Set

Our problem is now the following: find an algorithm whose input is a finite set of intervals S and whose output is a minimum generating set.

Motivation for the algorithm is contained in Fig. 3.1. First, recall Fig. 2.2. That figure shows that if G is a generating set containing $\{I, J_1, J_2, ..., J_k\}$ and $I = \bigcup_{i=1}^{k} J_i$, then $G - \{I\}$ is also a generating set. (Note: in general, if G generates S and G' generates G, then G' generates S.)

In Fig. 3.1, no single interval from $\{J_1, J_2, J_3, J_4, J_5\}$ can be deleted as in Fig. 2.2. However, if intervals $J_2 \cap J_3$ and $J_3 \cap J_4$ are added to G then J_2, J_3, and J_4 can all be deleted. By examining the corresponding rectangles in R, we see we have replaced a chain of three rectangles with a chain of two rectangles. The trick is successful because each point in $\bigcup_{i=1}^{5} J_i$ is contained in at least two intervals.

To construct the algorithm we first generalize this trick, obtaining a "reduction procedure." On input G, a dependent set of intervals, the procedure returns a new generating set with $|G| - 1$ intervals. The algorithm then iterates the procedure until G is independent.

To describe the reduction procedure formally, we first establish a few conventions. Recall that intervals can be viewed as a finite set of *atoms* or unit intervals $(i, i+1)$. (It is irrelevant whether intervals are open or

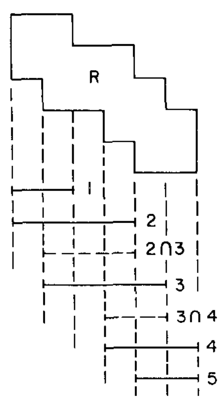


Fig. 3.1. Example of reduction procedure: $S = \{1, 2, 3, 4, 5\}$ is the set of intervals determined by R (Lemma 2.2). If $\{2 \cap 3, 3 \cap 4\}$ are added to S, then $\{2, 3, 4\}$ can be deleted as in Fig. 2.2. (Note: each interval i denoted by J_i in text.)

closed.) Let S be a set of intervals. Let $D=(L, R)$ be an interval (*not* necessarily a member of S). Then $S(D)=S(L, R)$ is the subset of intervals in S which are subintervals of (L, R) (i.e., the subset of intervals with both endpoints in (L, R)).

To help avoid confusion we shall call arbitrary intervals (L, R) *brackets*, and reserve the word "interval" for a member of S. Note, however that a bracket is an interval, not a set of intervals.

If $S=\{J_1, J_2,..., J_k\}$ then $\bigcup S=\bigcup_{i=1}^{k} J_i$ (the union of intervals in S). If $\bigcup S$ is a single bracket, then S is *connected.*

The following definitions are crucial to understanding the algorithm and the proofs of the next section.

DEFINITION 3.1 A connected subset of intervals $T \subseteq S$ is *simply dependent* if every atom in the bracket $\bigcup T$ is contained in at least two intervals in T. (Note: $T \subseteq S(\bigcup T)$.)

EXAMPLE 3.2. $S=\{(0, 3)\ (1, 3)\ (1, 4)\ (2, 4)\ (2, 5)\ (3, 5)\}$. $T_1=\{(1, 3)\ (1, 4)\ (2, 5)\ (3, 5)\}$ is simply dependent, as is $T_1 \cup \{(2, 4)\}$. $T_2= T_1 \cup \{(0, 3)\}$ is not simply dependent because the atom $(0, 1)$ is covered only by $(0, 3)$.

Alternatively, a simply dependent subset T contains two disjoint subsets of intervals, T_1 and T_2 with $\bigcup T_1=\bigcup T_2$. (A proof is in Section V.)

A simply dependent subset is a "smallest obstruction" to independence, as shown in the next lemma.

LEMMA 3.3. *A set T of intervals is independent if and only if T contains no simply dependent subset.*

Proof. Observe that a simply dependent set is also dependent (cannot be ordered as in Definition 2.4), but if T is independent, then every subset must be independent. Conversely, if T contains no simply dependent subset, then there is an atom x contained in exactly one interval I in T. Set $I_n=I$ (where $n=|T|$), then inductively order $T-\{I\}$. ∎

DEFINITION 3.4. A bracket $D=(L, R) \subseteq \bigcup S$ is *simply dependent in S* if the subset $S(D)$ is simply dependent.

(We shall also call D a *simple dependence*, when the set S is clear.)

In Example 3.2, $(1, 4)$, $(2, 5)$, and $(1, 5)$ are the only simple dependences: $S(1, 5)=T_1 \bigcup \{(2, 4)\}$. Note that $S(2, 4)=\{(2, 4)\}$ so $(2, 4)$ is *not* simply dependent. (Atoms $(2, 3)$ and $(3, 4)$ are each contained in two intervals in S but not in two intervals of $S(2, 4)$.)

Note that $\bigcup S$ contains a simply dependent bracket if and only if S contains a simply dependent subset.

The following definitions will be needed below to describe the minimum generating set algorithm.

DEFINITION 3.5. A bracket (L, R) is a *minimal simple dependence* if (L, R) is a simple dependence and no proper sub-bracket $(L', R') \subsetneqq (L, R)$ is a simple dependence.

EXAMPLE 3.6. Let $S = \{(0, 1)\ (0, 3)\ (0, 4)\ (1, 3)\ (1, 4)\ (3, 4)\}$. Then $(0, 4)$, $(0, 3)$, and $(1, 4)$ are all simply dependent in S, but only $(0, 3)$ and $(1, 4)$ are minimal. However, $(0, 4)$ is minimal in $S - \{(1, 3)\}$.

DEFINITION 3.7. If $T \subseteq S$, an interval $I \in T$ is *maximal* (in T) if no interval in T properly contains I.

In Example 3.6, $(0, 4)$ is the only maximal interval in S; $(0, 3)$ is maximal in $S(0, 3)$, but not in S. In Example 3.2 the maximal intervals in $S(1, 5)$ are $(1, 4)$ and $(2, 5)$.

Now assume $I_1, I_2, \ldots, I_k$ are the maximal intervals in a set T, ordered by left endpoints (which must be distinct). Note that they are also ordered strictly by right endpoints (proof by induction).

In Lemma 3.8 below we also need the following definition. A simple dependence (L, R) will be called *hinge-free*, if in $S(L, R)$, each pair of maximal intervals I_j, I_{j+1} which are adjacent in the ordering by left endpoints, intersect in at least one atom.

If (L, R) is not hinge-free then there is some pair which share an endpoint: $I_j = (X, Z)$ and $I_{j+1} = (Z, Y)$ (since every atom is contained in some maximal interval). The point Z separates (L, R) into two simply dependent brackets (L, Z) and (Z, R), where $S(L, R)$ is the disjoint union

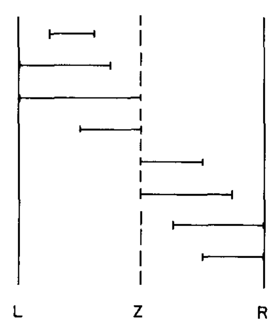


FIG. 3.2. Simple dependence (L, R) with a hinge Z; (L, Z) and (Z, R) are both minimal simply dependent.

of $S(L, Z)$ and $S(Z, R)$ (otherwise there would be a maximal interval between I_j and I_{j+1} in the ordering by left endpoints). Visually, the point Z appears as a mechanical hinge. (See Fig. 3.2.) Note that a minimal simple dependence (Definition 3.5) must be hinge-free.

We can now give the reduction procedure, *Reduce*, which is the main subroutine of the algorithm:

Reduce $[S, D]$

Input: S, a collection of intervals.
D, a simple dependence in S which is hinge-free.
Output: G, a generating set for S with $|S| - 1$ intervals.

(1) $T \leftarrow S(D)$ [subset of intervals contained in D as subintervals]
(2) Let $I_1, I_2, \ldots, I_k$ be the maximal intervals in T (definition 3.7), ordered by increasing left (and right) endpoints.
(3) **If** $k = 1$ **then**
$G \leftarrow S - \{I_1\}$
Else $G \leftarrow S \cup \{I_1 \cap I_2, I_2 \cap I_3, \ldots, I_{k-1} \cap I_k\} - \{I_1, I_2, \ldots, I_k\}$.
[Replace $I_1, I_2, \ldots, I_k$ with successive, pairwise intersections]
Exit

LEMMA 3.8. *When the reduction procedure is applied to S and D,*

(1) $|G| = |S| - 1$

(2) *G generates S*

(3) *if an atom x in D is covered by $m+1$ intervals in S, x is covered by m intervals in G.*

Proof. (1) follows because successive maximal intervals in $S(D)$ intersect in at least one atom, by the hinge-free asssumption.

To prove (2), notice that the only intervals in S which are not intervals in G are $I_1, I_2, \ldots, I_k$, so we need only verify that these are generated. For convenience, let I_0 and I_{k+1} be dummy (empty) intervals, so that G contains $I_j \cap I_{j+1}$ for $0 \leqslant j \leqslant k$. Now, for any $1 \leqslant j \leqslant k$ suppose that $I_{j-1} \cap I_j$ and $I_j \cap I_{j+1}$ do not generate I_j. The simple dependence of $S(D)$ guarantees that each atom in the interval $I_j - (I_{j-1} \cap I_j) - (I_j \cap I_{j+1})$ is contained in some interval in $S(D)$, which must be non-maximal, and hence contained in I_j. Thus, I_j is generated by intervals in G.

(3) follows because at least one maximal interval in $S(D)$ contains x; the effect of the reduction is to replace the $l+1$ maximal intervals containing x by l successive intersections. ▮

We cannot simply apply the reduction procedure to an arbitrary (hinge-free) simple dependence, continuing until the set is independent (i.e., contains no simple dependence). The following example shows that the generating set obtained may not be minimum.

EXAMPLE 3.9. $S = \{(0, 2)\ (0, 5)\ (1, 7)\ (3, 7)\ (4, 6)\ (4, 9)\ (5, 10)\ (8, 10)\}$;

$$D_1 = (0, 7) \qquad D_2 = (4, 10) \qquad D_3 = (0, 10)$$

(these are all simply dependent brackets).

Case. 1. Reduction procedure applied to D_3. Maximal intervals in $S(D_3)$ are $\{(0, 5)\ (1, 7)\ (4, 9)\ (5, 10)\}$:

$$G = \{(0, 2)\ (3, 7)\ (4, 6)\ (8, 10)\ (1, 5)\ (4, 7)\ (5, 9)\}.$$

Case 2. Reduction applied to D_1, then D_2:

$$G = \{(0, 2)\ (3, 7)\ (1, 5)\ (4, 6)\ (8, 10)\ (5, 9)\}.$$

In Case 1, $|G| = 7$, in Case 2, $|G| = 6$.

However, it turns out that if we choose D to be a *minimal* simple dependence (Definition 3.5), then the generating set is minimum.

ALGORITHM MGS.

Input: S, a finite set of intervals.
Output: G, a minimum generating set for S.

(1) $G \leftarrow S$
(2) **Do until** G contains no simple dependence:
  Let D be any minimal simple dependence in G
  $G \leftarrow$ *Reduce* $[G, D]$
 Exit

Observe that the algorithm must halt, since the cardinality of the generating set is reduced by one at each iteration. The generating set in Case 2 of Example 3.9 is one obtained by Algorithm MGS.

IV. VALIDITY OF ALGORITHM AND PROOF OF DUALITY THEOREM

In this section we prove that Algorithm MGS produces a minimum generating set. In essence, we give a recursive procedure for constructing an independent subset of S of size $|G|$, where G is any generating set produced by Algorithm MGS.

To state the theorems we need some notation. If S is any collection of intervals and D is a simply dependent bracket then RDS is the collection of intervals obtained by applying the reduction procedure to S and D. Think of RD as an operator, which is applied to a set of intervals.

We shall need three lemmas, which are stated below and proved in the Appendix. The lemmas help explain the need to choose a minimal simple dependence in Algorithm MGS.

LEMMA 4.1. *Let D be a minimal simple dependence in S. Let $B \subseteq \bigcup S$ be a bracket. If B is properly contained in D, then B is not simply dependent in RDS. If B is simply dependent in* RDS *then B is also simply dependent in S. (The reduction procedure creates no new simple dependence.)*

LEMMA 4.2. *Let D_1 and D_2 be distinct simply dependent brackets, such that neither contains the other as a sub-bracket. If D_1 is minimal then D_2 is also simply dependent in RD_1S. (The reduction can only destroy a dependence D_2 if $D_1 \subseteq D_2$.)*

LEMMA 4.3. *If D_1 and D_2 are distinct, minimal, simply dependent brackets in S, then the reduction procedure can be applied in either order, i.e., RD is commutative*:

$$RD_1 RD_2 S = RD_2 RD_1 S.$$

If D_1 $D_2,..., D_k$ is a sequence of brackets, let

$$S_k = RD_k RD_{k-1} \cdots RD_1 S$$

be the set of intervals obtained by successive applications of the reduction, so $S_k = \mathit{Reduce}[S_{k-1}, D_k]$. (We assume each D_i is simply dependent in S_{i-1} and hinge-free.) Call $D_1, D_2,..., D_k$ a *reduction sequence*.

We call a reduction sequence *optimal* if each D_i is minimal simply dependent in S_{i-1}. It is *complete* if S_k contains no simple dependence. We wish to show that any optimal, complete reduction sequence yields a minimum generating set. We first prove a lemma whose corollary allows us to rearrange optimal sequences, so that all brackets containing a given atom x come last in the sequence.

To prove the main theorem (4.6) choose an atom x covered once in S_k, then rearrange the reduction sequence produced by the algorithm. Delete all intervals containing x from S, to form S'. Delete all brackets containing x from the reduction sequence. It is shown that the truncated sequence is optimal for S'. Duality and simple arithmetic are used to complete the proof.

LEMMA 4.4. *Let S be a set of intervals. Let D_1 and D_2 be brackets such that*:

(a) *D_1 is minimal simply dependent in S;*

(b) *D_2 is minimal simply dependent in RD_1S;*

(c) *neither D_1 nor D_2 properly contains the other.*

Then:

(a′) *D_2 is minimal simply dependent in S;*

(b′) *D_1 is minimal simply dependent in RD_2S.*

Proof. Assume $D_1 \neq D_2$ (otherwise the result follows immediately) and that D_1 and D_2 overlap, otherwise the result is trivial.

(a′) By Lemma 4.1 and (c), D_2 is simply dependent in S. If D_2 were not minimal, so $D \subsetneqq D_2$ were simply dependent in S, then $D \not\subseteq D_1$ by (a), and $D_1 \not\subseteq D$ by (c). Then, D must be simply dependent in RD_1S by Lemma 4.2, contradicting (b).

(b′) D_1 is simply dependent in RD_2S by (c), (a′) and Lemma 4.2. As above, if $D \subsetneqq D_1$ were simply dependent in RD_2S, then by Lemma 4.1, $D \not\subseteq D_2$, and D must be simply dependent in S, contradicting (a). ∎

COROLLARY 4.5. *Let*

$$D_1,\ D_2,\ldots,\ D_i,\ D_{i+1},\ldots,\ D_k \tag{1}$$

be an optimal reduction sequence. Assume there is an atom x with $x \in D_i$, $x \notin D_{i+1}$. Then

$$D_1,\ D_2,\ldots,\ D_{i+1},\ D_i,\ldots,\ D_k \tag{1′}$$

is also optimal. Furthermore, the two sequences produce the same generating set S_k.

Proof. Let $S = S_{i-1}$, $D_1 = D_i$, $D_2 = D_{i+1}$. Then S, D_1, and D_2 satisfy (a), (b), and (c) of the lemma. Thus (1′) is optimal by (a′) and (b′).

By Lemma 4.3 (using (a) and (a′)), $S_{i+1} = RD_{i+1}RD_iS_{i-1} = RD_iRD_{i+1}S_{i-1}$, so the generating set S_k is not affected by exchanging D_i and D_{i+1}. ∎

THEOREM 4.6. *Let S be a set of intervals. Let $D_1, D_2,\ldots, D_m$ be any optimal, complete reduction sequence for S ($m \geqslant 0$). (So S_m is a possible output of Algorithm* MGS.) *Then S contains an independent subset T with*

$$|T| = |S| - m = |S_m|.$$

Proof. By induction on $|S|$. $|S| = 0$ is trivial. Also, $m = 0$ is trivial (for all $|S|$) since then S itself is independent.

Recall, $S_k = RD_kRD_{k-1} \cdots RD_1 S$ for all k. Since S_m contains no simple dependence, there is an atom x contained in exactly one interval of S_m, and in at least one interval of S.

Let $S\{x\}$ denote the subset of intervals in S which contain x. Let $S' = S - S\{x\}$. We shall apply induction to S'.

By repeated application of Corollary 4.5 we may assume either:

(i) no D_j contains x; or,

(ii) there is an index $0 \leqslant i < m$ such that $D_{i+1}, D_{i+2},..., D_m$ contain x and $D_1, D_2,..., D_i$ do not contain x.

Observe that no interval in $S\{x\}$ is in $S(D_j)$ for $1 \leqslant j \leqslant i$, so $D_1, D_2,..., D_i$ is also optimal for S'. We claim it is also complete. Otherwise, there is a simple dependence D in $S_i' = RD_iRD_{i-1} \cdots RD_1 S'$. However, $S_i = S_i' \cup S\{x\}$ so D is also simply dependent in S_i. Since $x \notin D$, $D_j \nsubseteq D$ for $i+1 \leqslant j \leqslant m$; by repeated application of minimality and Lemma 4.2, D is simply dependent in S_m, contradicting the independence of S_m.

By induction, there is an independent set $T' \subseteq S'$ with

$$|T'| = |S'| - i.$$

For any $I \in S\{x\}$, $T = T' \cup \{I\}$ is independent. By Lemma 3.8(3)

$$|S\{x\}| = m - i + 1.$$

Therefore,

$$\begin{aligned} |T| = |T'| + 1 &= |S'| - i + 1 \\ &= |S| - (m - i + 1) - i + 1 \\ &= |S| - m. \qquad \blacksquare \end{aligned}$$

Since $|T| \leqslant |G|$ for any generating set, Algorithm MGS produces a minimum generating set. We have proved the duality theorem for intervals:

Corollary 4.7 (Györi). *For any set of intervals S*

$$\alpha(S) = \theta(S). \quad \blacksquare$$

Remark. If G' is a minimum generating set for S' (in Theorem 4.6), it is not necessarily true that there is some interval J such that $G' \cup \{J\}$ generates S. For example, let $S = \{(0, 3)\ (0, 5)\ (2, 6)\ (4, 6)\}$, $G = S_1 = \{0, 3)\ (2, 5)\ (4, 6)\}$ and let $x = (0, 1)$. Then $S' = \{(2, 6)\ (4, 6)\} = G'$, and no single interval can be added which generates $(0, 3)$ and $(0, 5)$.

V. Implementation

We now outline how to implement Algorithm MGS in polynomial time. We shall see that algorithm is at worst quadratic in n, the number of intervals, and v, the number of vertices. Represent intervals by ordered pairs (X, Y) of integers. Assume the intervals are stored in a doubly linked list, ordered by increasing left endpoints. In case of ties, the order is arbitrary. We also assume we have an ordered list of all endpoints, marked as to whether they are left, right, or both.

Assume that S, the set of intervals, is connected, i.e., $\bigcup S$ is a single bracket. Otherwise find generating sets for each connected component separately.

Recall that the algorithm to find a minimum generating set has two basic steps: (1) given S, the collection of intervals, find (L, R), a minimal simply dependent bracket in $\bigcup S$; (2) apply the reduction procedure to (L, R) and update S.

The reduction procedure is easily accomplished in linear time. If (X_i, Y_i) denotes the ith maximal interval in $S(L, R)$ in the ordering by left endpoints, then $(X_{i+1}\ Y_{i+1})$ is the leftmost interval with $X_i < X_{i+1} < Y_i < Y_{i+1} \leqslant R$, and such that (X_{i+1}, Y_{i+1}) is the longest interval in (L, R) with left end X_{i+1}. Here is a more formal description of the procedure:

Reduce $[S, L, R)]$

[Replaces maximal intervals in $S(L, R)$ by successive intersections]

(1) $(X_1, Y_1) \leftarrow$ longest interval in $S(L, R)$ with left end L.

(2) **While** $Y_1 < R$ **do**

  Let X be the leftmost endpoint with an interval (X, Y) satisfying $X_1 < X < Y_1 < Y \leqslant R$

(3) $(X_2, Y_2) \leftarrow$ longest such interval with $X_2 = X$

(4) $S \leftarrow S - \{(X_1, Y_1)\} \cup \{(X_2, Y_1)\}$

   [Delete I_1 and add $I_1 \cap I_2$]

(5) $(X_1, Y_1) \leftarrow (X_2, Y_2)$

 Endwhile

(6) [$Y_1 = R$ if (L, R) is simply dependent]

   $S \leftarrow S - \{(X_1, Y_1)\}$

   [Delete rightmost maximal interval]

   Reduce $[S, L, R] \leftarrow S$

 Exit

A minimal simple dependence (L, R) can be found in linear time using the following idea. Beginning at the right end R, simultaneously construct two disjoint "paths" (connected subsets) of intervals from R. Extend paths by adding intervals. To ensure that we find a minimal simple dependence, always extend the shorter path first, and always choose the interval which extends the path by the smallest nonzero amount. This is the idea behind the following subroutine.

Procedure SD $[S, R, X_1, X_2]$

[Finds rightmost L such that (L, R) is simply dependent.]

(1) Unmark all intervals

(2) [We assume (X_1, R) and (X_2, R) are the shortest two intervals with right end R, and $X_1 \neq X_2$.]

   $L_1 \leftarrow X_1$ $L_2 \leftarrow X_2$ [Initialize path lengths]

(3) Mark (X_1, R) and (X_2, R) [Add (X_1, R) to path 1 and (X_2, R) to path 2]

(4) **While** $L_1 \neq L_2$ **do**

   Let L_i be the rightmost of $\{L_1, L_2\}$ [path i is shorter]

(5) **If** there is an unmarked interval (X, Y) with $X < L_i \leqslant Y \leqslant R$

   Then $L_i \leftarrow$ rightmost such X [Extend path i by smallest amount]

   Mark corresponding interval (X, Y) [Add (X, Y) to path i]

   Else *Procedure SD* $[S, R, X_1, X_2] \leftarrow \varnothing$

      Exit [No simple dependence (L, R)]

Endwhile

(6) *Procedure SD* $[S, R, X_1, X_2] \leftarrow L_1$ $[L_1 = L_2]$

Exit $[(L_1, R)$ is a minimal simple dependence$]$

To see that Procedure *SD* works, first observe that it must halt, since S is finite. Suppose it halts having completed the while loop $j \geqslant 1$ times; let $L_1(j)$ and $L_2(j)$ be the corresponding values of L_1 and L_2. If $L_1(j) = L_2(j)$ then, since the paths are disjoint, $(L_1(j), R)$ is simply dependent. Thus, if there is no simple dependence (L, R), the procedure has output $\varnothing$. If (L', R) is the minimal simple dependence with right end R, then $L_1(j) \leqslant L'$ ($L_1(j)$ is weakly left of L').

Note that $L' \leqslant L_1(1)$ and $L' \leqslant L_2(1)$. Assume by induction that $L' \leqslant L_1(i)$ and $L' \leqslant L_2(i)$ for $1 \leqslant i \leqslant j-1$. We have assumed $L_1(j-1) \neq L_2(j-1)$, so there is an atom x in (L', R) covered by only one marked interval. But the simple dependence of (L', R) guarantees that an umarked $(X, Y) \in S(L', R)$ containing x exists. Such an interval will be chosen in step (5) because the rightmost X is chosen. So we must have

$$L' \leqslant L_1(j) = L_2(j) \leqslant L'.$$

We can now put the pieces together to construct the main algorithm. Suppose we know $R_1 < R_2 < \cdots R_k$, all the right endpoints of intervals. For each i, let c_i be the number of intervals with right end i; then we need test only those R_i for which $c_i \geqslant 2$. If $c_i \geqslant 2$, then we need determine the two shortest intervals with right end R_i only once: when we reduce a simply dependent bracket (L_i, R_i), of the intervals with right end R_i, only the longest is removed, which does not affect the order of those remaining.

ALGORITHM MGS [S]

[Constructs minimum generating set for S]

(1) Let $R_1 < R_2 < \cdots < R_k$ be all right endpoints of intervals, ordered from left to right.

  For $i = 1$ to k **do**

    $c_i \leftarrow$ number of intervals with right end R

    $X_{i1}, X_{i2} \leftarrow$ left endpoints of shortest two intervals with right end R [Possibly $X_{i2} = \varnothing$]

  Endfor

(2) **For** $i = 1$ to k **do**

(3)   **While** $c_i \geqslant 2$ **do** [R_i is a candidate right end]

    $L_i \leftarrow$ *Procedure SD*, [S, R_i, X_{i1}, X_{i2}]

    [(L_i, R_i) is minimal simply dependent]

(4)   **If** $L_i \neq \varnothing$ **then** $S \leftarrow$ *Reduce* [S, L_i, R_i]

     $c_i \leftarrow c_i - 1$

(5)   **Else** $c_i \leftarrow 0$ [No simple dependence with right end R_i]

(6)   **Endwhile**

(7) **Endfor**

  Algorithm MGS [S] $\leftarrow S$

  Exit

To show Algorithm MGS is correct we must only show that if $c_1, c_2, \ldots, c_{i-1} \leqslant 1$ (after $i-1$ passes through step (2)) then the reduction of (L_i, R_i) creates no simple dependence $D = (L, R)$ with R to the left of R_i. But, this follows from Lemma 4.1, since $c_j \leqslant 1$ implies there is no simple dependence with right end R_j.

Since the intervals are stored in a left-ordered list, *Reduce* and *Procedure SD* are at worst linear in n, the number of intervals. Step (1) of Algorithm MGS may be performed, using our original list of all endpoints, by reading through the list of intervals once, simultaneously keeping a tally of intervals and storing the left endpoints of the shortest two intervals found for each right end. This procedure is $O(n \log k)$, where k is the number of right ends. If k_1 is the number of times Procedure *SD* is performed, and m is the number of times the reduction is performed, the algorithm is $O((\log k + k_1 + m)\, n)$, which is $O(n^2)$ in the worst case.

It is not difficult to construct a vertically convex polygon such that this implementation of MGS achieves the worst case running time for the corresponding set of intervals.

To see that the number of intervals is at most v, the number of vertices of the polygon, observe that the topmost horizontal slice for each interval contains at least one corner (vertex) of the polygon along the top edge of the slice. These corners are distinct, so $n \leqslant v$. (One can actually show, using

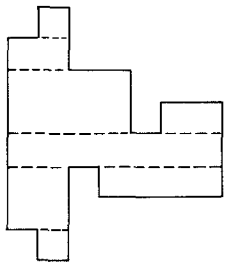


FIG. 5.1. Partitioning with horizontal chords.

results of Ohtsuki (1982), that for vertically convex polygons, $v-4/8 \leqslant n \leqslant v/2$.)

In order to complete the analysis of the algorithm for constructing a rectangle cover, we must also consider: (1) constructing the set of intervals determined by a vertically convex polygon; and (2) reconstructing the rectangle cover from the interval generating set. (We shall outline the solution, but omit the details.) (1) can be accomplished by first partitioning the polygon using horizontal chords only (see Fig. 5.1), as in Ohtsuki (1982); the intervals determined by these rectangles are those determined by inequivalent slices. The partitioning can be accomplished easily in $O(v^2)$ steps, or in $O(v \log v)$ steps using a balanced tree to store vertices.

To solve (2), first observe that since the polygon R is vertically convex, to each atom there corresponds exactly one upper and one lower horizontal boundary segment of R. Thus, for each interval we need to find the minimum upper segment and maximum lower segment among those corresponding to the atoms covered by the interval. Assuming the polygon grid is "normalized" so that each grid line contains some vertex, then solving (2) is trivially $O(v^2)$. By using a group-tree structure to store intervals (due to McCreight, in Ullman, 1984, pp. 384–391) we can actually construct the rectangles in $O(v \log v)$ steps.

VI. Open Questions

We have given an efficient algorithm for constructing a minimum rectangle cover for a class of polygons which are essentially 1-dimensional. Franzblau (1985) has discovered recently that an analogous duality theorem and minimum cover construction hold for k-dimensional rectilinear polygons that are "$k-1$ convex." For arbitrary, simply connected rectilinear polygons there is neither a simple reduction to an interval problem, nor a duality theorem. However, it is possible that a generalization of the interval reduction algorithm will produce a minimum cover for some larger class of non-convex polygons. We have studied the problem of covering regions which can be partitioned into a few vertically convex components, but even this simple extension has proved quite difficult to analyze. Preliminary results suggest that finding a minimum cover for any simply connected polygon is *NP*-hard. Traditional "component design" proof techniques seem to be useless here.

We would also like to know whether one of the heuristics we have mentioned, or another (such as using our reduction procedure for "local improvement") has a constant performance ratio. We conjecture that the answer is yes.

Another interesting algorithmic question is rectangle covering for polygons with diagonal (45°) edges. (This is a more realistic approximation for integrated circuit applications.) Assume that polygons have integer vertices and no acute angles, and that rectangles can have two orientations. Is there again a restricted class of such polygons (e.g.) convex in two directions) with an analogous duality theorem? What are good heuristics in this case? These questions are also open for 3-dimensional polygons.

Finally, our reduction procedure is similar in character to augmenting path methods for maximum matching (Lawler, 1976), but neither our result, nor the duality theorem seem to be a consequence of these classical duality results. However, it is likely that there is some deeper connection between these theorems and algorithms.

Appendix: Proofs of Lemmas of Section IV

Lemma 4.1. *Let D be a minimal simple dependence in S. Let $B \subseteq \bigcup S$ be a bracket. If B is properly contained in D then B is not simply dependent in RDS. If B is simply dependent in RDS then B is simply dependent in S.*

Proof. First assume B is properly contained in D. Let $D=(L, R)$ and $B=(X, Y)$. Let $I_j, I_{j+1},..., I_l$ be the ordered maximal intervals in $S(D)$ such that $I_j \cap I_{j+1}, I_{j+1} \cap I_{j+2},..., I_{l-1} \cap I_l$ are in $RDS(B)$. Then $I_{j+1}, I_{j+2},..., I_l$ are in $S(X, R)$ and $I_j, I_{j+1},..., I_{l-1}$ are in $S(L, Y)$. Suppose every atom in (X, Y) were covered twice in $RDS(X, Y)$ but not in $S(X, Y)$; then, by the minimality of D, there is at least one pair I_j, I_{j+1}. Since these are adjacent maximal intervals, no interval in $S(D)$ contains atoms in both (L, X) and (Y, R). Thus, every atom in (L, Y) is covered twice in $S(L, Y)$ and every atom in (X, R) is covered twice in $S(X, R)$, which violates the minimality of (L, R).

The remainder of the proof is trivial if B and D do not intersect, or $D \subseteq B$. Thus assume $(L, R)=D$ and $(X, Y)=B$ intersect in the bracket (X, R). (The same argument works when the intersection is (L, Y).) If

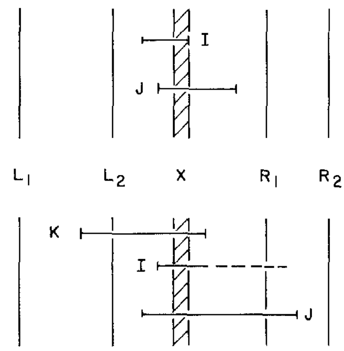


FIG. A.1. Configurations in proof of Lemma 4.2 (with atom x shaded). Top: $I, J \in S(D_1 \cap D_2)$; bottom: $J \notin S(D_1 \cap D_2)$.

$I_j \cap I_{j+1}$ is in (X, R), where I_j and I_{j+1} are adjacent maximal intervals in $S(L, R)$, then I_{j+1} is in (X, R). Thus, any atom covered twice in $RD(X, Y)$ is covered twice in $S(X, Y)$ as well. ∎

LEMMA 4.2. *Let D_1 and D_2 be distinct simply dependent brackets in S, such that neither contains the other as a sub-bracket. If D_1 is minimal then D_2 is simply dependent in $RD_1 S$.*

Proof. Let $D_1 = (L_1, R_1)$ and $D_2 = (L_2, R_2)$. We may assume D_1 is left of D_2: $L_1 < L_2 < R_1 < R_2$ (if D_1 and D_2 do not overlap the result is trivial). Then, $D_1 \cap D_2 = (L_2, R_1)$.

Assume D_2 is not simply dependent in $RD_1 S$. Then there is an atom x in $D_1 \cap D_2$ covered by at most one interval in $RD_1 S(D_2)$. However, the simple dependence of D_2 means that x is contained in exactly two intervals I, J in $S(D_2)$ (see Fig. A.1.).

Suppose I and J are both contained in $S(D_1 \cap D_2)$. As a corollary of Lemma 3.8(3) we may assume no other interval in $S(D_1 \cap D_2)$ contains x. Suppose some other interval in $S(D_1)$ contains x. By verifying the three cases (I, J not maximal in $S(D_1)$; I maximal only; I, J both maximal) one finds a contradiction: there are two intervals in $RD_1 S(D_1 \cap D_2)$ containing x. So, we may assume I, J are the only intervals in $S(D_1)$ containing x.

However, by the simple dependence of $S(D_2)$, all atoms left of $x = (i, i+1)$ in (L_2, R_1) are covered twice in $S(L_2, i) \cup \{I, J\}$. By the simple dependence of $S(D_1)$, all atoms right of x are covered twice in $\{I, J\} \cup S(i+1, R_1)$. But then $S(L_2, R_1) = S(D_1 \cap D_2)$ is simply dependent, violating the minimality of D_1.

Now suppose $J \in S(D_2)$ but $J \notin S(D_1 \cap D_2)$. The simple dependence of $S(D_1)$ means there is an interval $K \in S(D_1)$ containing x with $K \notin S(D_1 \cap D_2)$. Thus, contrary to our assumption, in $RD_1 S(D_2)$ there are

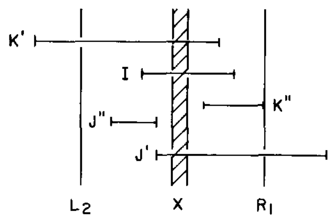


FIG. A.2. Configuration in proof of Lemma 4.3.

two intervals containing x: J, and either I or the intersection of I with some (maximal) interval in $S(D_1)$ containing x. ∎

LEMMA 3. *If D_1 and D_2 are distinct minimal simply dependent brackets in $\bigcup S$, then*

$$RD_1 RD_2 S = RD_2 RD_1 S.$$

Proof. As in Lemma 4.2 we may assume $L_1 < L_2 < R_1 < R_2$. We may further assume that $S(D_1 \cap D_2)$ is nonempty, otherwise $S(D_1) = RD_2 S(D_1)$ and $S(D_2) = RD_1 S(D_2)$, and the result follows directly. The minimality of D_1 (or D_2) implies that there is some atom x contained in exactly one interval I in $S(D_1 \cap D_2)$. The simple dependence of D_1 and D_2 means there exist intervals $J \in S(D_2)$ and $K \in S(D_1)$ which contain x.

Any maximal interval in $S(D_2)$ to the left of x is contained in K; any maximal interval in $S(D_1)$ to the right of x is contained in J; and I is the only interval in $S(D_1 \cap D_2)$ containing x. Thus, I is the only interval which can be maximal in both $S(D_1)$ and $S(D_2)$.

If I is not maximal in $S(D_1)$ (e.g.) then all the "new" intervals in $RD_1 S(D_1)$ (intersections of maximal intervals in $S(D_1)$) are either not in $D_1 \cap D_2$ or are contained in J. Thus, the maximal intervals in $RD_1 S(D_2)$ are the same as those in $S(D_2)$. On the other hand, the new intervals in $RD_2 S(D_2)$ are either not in $D_1 \cap D_2$, are contained in K, or are contained in I. Again, the maximal intervals in $RD_2 S(D_1)$ are the maximal intervals in $S(D_1)$. Therefore $RD_1 RD_2 S = RD_1 RD_2 S$, if I is not maximal in both $S(D_1)$ and $S(D_2)$.

If I is maximal in both sets, we must verify that the same list of intervals is produced in either order. If the reducton is applied to D_1 first, an argument similar to the preceding case shows that the maximal intervals in $RD_1 S(D_2)$ are the same as in $S(D_2)$ except that I is replaced by $I \cap K'$. K' is the maximal interval in $S(D_1)$ adjacent to, and to the left of I. (K' contains x because K contains x, see Fig. A.2.) If D_2 is reduced first, the maximal intervals in $RD_2 S(D_1)$ are the same as in $S(D_1)$ except that I is replaced by $I \cap J'$, where J' is the maximal interval in $S(D_2)$ adjacent to and to the right of I.

Let J'' be the maximal interval in $S(D_2)$ adjacent to and left of I, and K'' the maximal interval in $S(D_1)$ adjacent to and right of I (either J'' or K'' may not exist). (See Fig. A.2.) To verify that we obtain the same intervals in either order, first observe that $K' \cap I \cap J'$ is always non-empty and is produced during the second reduction. If D_1 is reduced first then $J'' \cap K' \cap I = J'' \cap I$ is produced on the second reduction, while $I \cap K''$ is produced on the first. If D_2 is reduced first then $I \cap J' \cap K'' = I \cap K''$ is produced second while $I \cap J''$ is produced first. (If $J'' = \emptyset$ or $K'' = \emptyset$, this is still true.) ▮

ACKNOWLEDGMENTS

We thank one of the referees in particular for giving an excellent, perceptive critique, which led to a much improved paper, and for bringing several references to our attention.

RECEIVED May 31, 1984; REVISED February 15, 1985

REFERENCES

AIGNER, M. (1979), "Combinatorial Theory," Chaps. VI–VIII, Springer–Verlag, Berlin/New York.

ALBERTSON, M. O. AND O'KEEFE, C. J. (1981), Covering regions with squares, *SIAM J. Algebraic Discrete Methods* **2** No. 3, 240–243.

BERGE, C., CHEN, C. C., CHVÁTAL, V., AND SEOW, C. S. (1981), Combinatorial properties of polyominoes, *Combinatorica* **1** No. 3, 217–224.

CHAIKEN, S., KLEITMAN, D. J., SAKS, M., AND SHEARER, J. (1981), Covering regions by rectangles, *SIAM J. Algebraic Discrete Methods* **2**, 394–410.

CHVÁTAL, V. (1979), A greedy heuristic for the set-covering problem, *Math. Oper. Res.* **4** No. 3, 233–235.

FRANZBLAU, D. S. (1985), unpublished manuscript, MIT, Cambridge, Mass.

FRANZBLAU, D. S., AND KLEITMAN, D. J. (1984), An algorithm for constructing regions with rectangles, *in* "16th ACM Sympos. on Theory of Computing," pp. 167–174.

GAREY, M. S., AND JOHNSON, D. S. (1979), "Computers and Intractability: A Guide to the Theory of *NP*-Completeness," Freeman, San Francisco.

GYÖRI, E. (1984), A minimax theorem on intervals, *J. Combin. Theory Ser. B* **37** No. 1, 1–9.

HEGEDÜS, A. (1982), Algorithms for covering polygons by rectangles, *Computer Aided Design* **14** No. 5, 257–260.

JOHNSON, D. S. (1974), Approximation algorithms for combinatorial problems, *J. Comput. System Sci.* **9**, 256–278.

JOHNSON, D. S. (1982), The *NP*-completeness column: An ongoing guide, *J. Algorithms* **3**, 182–195.

LAWLER, E. L. (1976), "Combinatorial Optimization: Networks and Matroids," Holt, Rinehart, & Winston, New York.

LEVCOPOULOS, C., AND LINGAS, A. (1984), Covering polygons with minimum number of rectangles, *in* "Proc. Sympos. on Theoret. Aspects of Comput. Sci. (STACS 84), *in* "Lect. Notes in Comput. Sci. Vol. 166, pp. 63–72, Springer–Verlag, Berlin/New York.

MASEK, W. J. (1978), Some *NP*-complete set covering problems, unpublished manuscript, MIT (Referenced in Garey and Johnson, 1979, p. 232).

MEAD, C., AND CONWAY, L. (1980), "Introduction to VLSI Systems," Chaps. 2 and 4, Addison–Wesley, Reading, Mass.

OHTSUKI, T. (1982), Minimum dissection of rectilinear regions, *in* "Proc. 1982 IEEE Sympos. on Circuits and Systems," Rome, pp. 1210–1213.

O'ROURKE, J. (1982), The complexity of computing minimum convex covers for polygons, *in* "Proc. 20th Allerton Conf.," Allerton, Ill., Oct., pp. 75–84.

O'ROURKE, J., AND SUPOWIT, K. J. (1983), Some *NP*-hard polygon decomposition problems, *IEEE Trans. Inform. Theory* **29**, No. 2, 181–190.

PAGLI, L., LODI, E., LUCCIO, F., MUGNAI, C., AND LIPSKI, W. (1979), On two-dimensional data organization 2, *Fund. Inform.* **2**, 211–226.

SAKS, M. (1982), A class of perfect graphs associated with planar rectilinear regions, *SIAM J. Algebraic Discrete Methods* **3** No. 3, 330–342.

ULLMAN, J. D. (1984), "Computational Aspects of VLSI," Chap. 9, Comput. Sci., Rockville, Md.