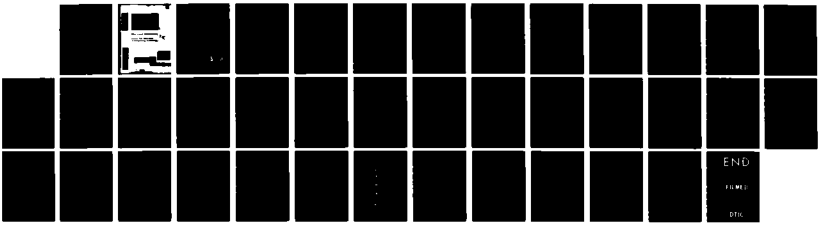
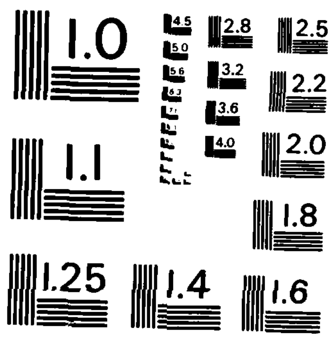
AD-A161 378 COMPLEXITY OF THE GENERALIZED MOVER'S PROBLEM(U) HARVARD UNIV CAMBRIDGE MA AIKEN COMPUTATION LAB J H REIF 1985 TR-04-85 N00014-80-C-0647 1/1

UNCLASSIFIED F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

AD-A161 378

COMPLEXITY OF THE GENERALIZED

MOVER'S PROBLEM

John H. Reif

TR-04-85

Harvard University

Center for Research in Computing Technology

Aiken Computation Laboratory
33 Oxford Street
Cambridge, Massachusetts 02138

COMPLEXITY OF THE GENERALIZED

MOVER'S PROBLEM

John H. Reif

TR-04-85

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. A[illegible] A161[illegible]

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*)

COMPLEXITY OF THE GENERALIZED MOVER'S PROBLEM

5. TYPE OF REPORT & PERIOD COVERED

Technical Report

6. PERFORMING ORG. REPORT NUMBER

TR-04-85

7. AUTHOR(*s*)

John H. Reif

8. CONTRACT OR GRANT NUMBER(*s*)

N00014-80-C-0647

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Harvard University
Cambridge, MA 02138

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

12. REPORT DATE

1985

13. NUMBER OF PAGES

35

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)

Same as above

15. SECURITY CLASS. (*of this report*)

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

unlimited

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

robotics, mover's problem, obstacle avoidence, PSPACE, combinatorial, geometry, collision avoidence

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

See reverse side.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

ABSTRACT

This paper concerns the problem of planning a sequence of movements of linked polyhedra through 3 dimensional Euclidean space, avoiding contact with a fixed set of polyhedra obstacles. We prove this generalized mover's problem is polynomial space hard. Our proof provides strong evidence that robot movement planning is computationally intractable, i.e., any algorithm requires time growing exponentially with the number of degrees of freedom.

Complexity of the Generalized Mover's Problem[+]

John H. Reif*

Aiken Computation Lab.
Division of Applied Sciences
Harvard University
Cambridge, Massachusetts 02138

*This work was supported by the Office of Naval Research Contract N00014-80-C-0647.

+Also to appear in "Planning, Geometry and Complexity of Robot Motion", ed. by Jacob Schwartz, Ablex Publ., Norwood, NJ, 1985.

ABSTRACT

This paper concerns the problem of planning a sequence of movements of linked polyhedra through 3 dimensional Euclidean space, avoiding contact with a fixed set of polyhedra obstacles. We prove this generalized mover's problem is polynomial space hard. Our proof provides strong evidence that robot movement planning is computationally intractable, i.e., any algorithm requires time growing exponentially with the number of degrees of freedom.

1. INTRODUCTION

1.1 The Movers Problem

The classical *mover's problem* in d-dimensional Euclidean space is:

Input: (O,P,p_I,p_F) where O is a set of polyhedral *obstacles* fixed in Euclidean space and P is a rigid polyhedron with distinguished initial position p_I and final position p_F. The inputs are assumed to be specified by systems of rational linear inequalities.

Problem: Can P be moved by a sequence of translations and rotations from p_I to p_F without contacting any obstacle in O?

For example, P might be a sofa* which we wish to move through a room crowded with obstacles. Figure 1 gives a simple example of a two dimensional movers problem.

The mover's problem may be *generalized* to allow P (the object to be moved) to consist of multiple polyhedra freely linked together at various distinguished vertices. (A typical example is a robot arm with multiple joints.) Again, the input is specified by systems of rational linear inequalities. (A precise definition of the generalized problem is given in Section 2.)

1.2 Lower Bounds for Generalized Mover's Problems

Our main result, first presented in [Reif, 79] (and given in full detail in Section 2) is that the generalized mover's problem in three dimensions is

*The author first realized the nontrivial mathematical nature of this problem when he had to plan the physical movement of an antique sofa from Rochester to Cambridge.

polynomial space hard. That is, we prove that the generalized mover's problem is at least as hard as any computational problem requiring polynomial space. (Polynomial space problems are at least as hard as the well known NP problems; see [Garey and Johnson, 79].)

This was the first paper investigating the inherent computational complexity of a robotics problem, and in fact was the first polynomial space hardness result for any problem in Computational Geometry. Our proof technique is to use the degrees of freedom of P to encode the configuration of a polynomial space bounded Turing maching M, and to design obstacles which forced the movement of P to simulate the computation of M.

This work was originally motivated by applications to robotics: the author felt it was important to examine *computational complexity issues in robots* given the recent development of mechanical devices autonomously controlled by micro and minicomputers, and the swiftly increasing computational power of these controllers. However, it took a number of years before computational complexity issues in robotics became of more general interest. Recently there have been a flurry of papers in the now emerging area which we might term *Computational Robotics*.

Recent investigations in lower bounds have provided some quite ingenious lower bound constructions for restricted cases of the generalized mover's problem. For example, [Hopcroft, Joseph, and Whitesides, 82] showed that the generalized mover's problem in three dimensions is also polynomial space hard, and [Hopcroft and Sharir, 84] show that the problem of moving a collection of disconnected polyhedra in a two dimensional maze is polynomial space hard. The problem of moving a collection of disks in two dimensions is known to be NP-hard [Sparakis and Yap, 85], but is remains open to show this problem polynomial space hard.

1.3 Upper Bounds for Mover's Problems

Our lower bounds for the generalized mover's problem provide evidence that time bounds for algorithms for movement planning must grow exponentially with the number of degrees of freedom. We next give a brief discription of known algorithms for mover's problems. In our original paper [Reif, 79] we also sketched a method for efficient solution of the classic mover's problem where P, the object to be moved, is rigid. In spite of considerable work on this problem by workers in the robotics fields and in artificial intellegence, (for example [Nilson, 69], [Paul, 72], (Udupa, 77], [Widdoes, 74], [Lozano-Pérez and Wesley, 79]) no algorithm guaranteed to run in polynomial time had previously appeared. Our approach was to transform a classic mover's problem (O,P,p_I,p_F) of size n in d dimensions to an apparently simpler mover's problem (O',P',p_I',p_F') of dimension d', where P' is a single part and d' is the number of degrees of freedom of movement in the original problem. The transformed problem is thus to find a path in d'-dimensional space avoiding the transformed obstacles O. The fundamental difficulty is that the induced obstacles may be non-linear constraints. ([Lozano-Pérez and Wesley, 79] did not construct O', but instead approximated the induced obstacles O' by linear constraints. Unfortunately, an exponential number of linear constraints were required to approximate even a quadratic constraint within accuracy 2^{-n}. Thus their method required exponential time (i.e., 2^{cn} time for some $c > 0$) even if the original mover's problem was two dimensional.)

<u>Example</u>. Consider a classical mover's problem (O,P,p_I,p_F) restricted to *dimension* $d = 2$, with the obstacles O consisting of a set of line segments and P a single polygon. A *position* of P can be specified by a triple (x,y,θ) where (x,y) are the cartesian coordinates of some fixed vertex of P and θ is the angle of rotation around this vertex. We define

a mapping f from the position of P to 3-space. Let $f(x,y,\theta) = (x',y',z')$ where $y = z'$, $\tan(\theta) = x'/y'$, and $x = (x')^2 + (y')^2 - \alpha$, for some sufficiently large constant $\alpha \geqslant 0$. (α may be taken as the diameter of a circle enclosing P.) See Figure 3.

In this case, we define a *1-contact set* to be a maximal set of positions of P where a vertex of P contacts a line segment of O, or a vertex of $\bar{O}$ contacts a line segment of P. (See Figure 4.) The transformed obstacles O' are the union of these 1-contact sets. Thus each obstacle in O' is a quadratic surface patch which may be easily constructed from the input, there are at most $O(|O||P|)$ such obstacles and their $O(|O|^2|P|^2)$ intersections can easily be computed within accuracy $2^{n^{-c}}$ for any $c > 0$, by known polynomial time procedures [Comba, 68] for intersection of quadratic surface patches. Hence in this simple example the connected regions bounded by O' can be explicitly constructed in polynomial time within accuracy 2^{-n^c} which is sufficient for solution of this mover's problem.

In the case of a classical mover's problem (O,P,p_I,p_F) of dimension $d = 3$, the transformed problem (O',P',p_I',p_F') has dimension $d' = 6$. In this case we define a 1-contact set to be a maximal set of positions of P where an edge of P contacts a face of O or an edge of O contacts a face of P. Again, the 1-contact sets are constant degree polynomials. The transformed obstacles O' are the union of the 1-contact sets. The connected regions defined by O' can again be explicitly constructed by intersecting these constraints. In [Reif, 79], we briefly suggested a method for this construction, but the full credit should be given to [Schwartz and Sharir, 83A] who later gave a complete detailed description of a method for explicit construction of such a transformed movers problem in 3 dimensions in polynomial time. ([O'Dunlaing, Sharir, and Yap, 83] further improved this construction by observing that movement of P can be restricted to be equidistant from the obstacles.)

This approach was extended by [Schwartz and Sharir, 83B] to solve any generalized mover's problem of input size n with d' degrees of freedom in time $n^{2^{O(d')}}$. They make use of the algebraic decomposition of [Collins, 75] (previously used to decide formulas of the theory of real closed fields) to construct the connected regions bounded by O'. Note that their upper bounds grow doubly exponentially with d', where as our polynomial space lower bounds suggest only single exponential time growth with d'. It remains a challenging problem to close the gap between those lower and upper bounds for generalized movers problems. Further progress will likely depend on improvements to decision algorithms for the theory of real closed fields; recently [Ben-Or, Kozen, and Reif, 84] gave a single exponential space decision algorithm.

1.4 Further Problems in Computational Robotics

There are some very challenging problems remaining in the field of Computational Robotics beyond the complexity of the mover's problem and its generalization. We mention below three such problems and some recent progress.

(1) Frictional Movement. The problem here is to plan movement for (O,P,p_I,p_F) in the case contact is allowed in the presence of friction between surfaces. [Rajan and Schwartz, 85] gives the first known decision algorithm in the case that O is a cylindrical hole and P is a peg. [Miller and Reif, 85] prove undecidability of planning frictional movement. What natural subclass of frictional movement problems is decidable?

(2) Minimal Movement. The problem is, given a set of k polygonal obstacles in d space defined by a total of n linear constraints, and points p_I, p_F find a minimal length path from p_I to p_F avoiding the obstacle O. [Chazelle, 82] gives a $O(n \log n)$ algorithm in the case $d=2$ and $k=1$. [Sharir and Schorr, 84] give a $2^{2^{O(n)}}$ algorithm for $d=3$.

Recently [Reif and Storer, 85] gave a $O(nk \log n)$ algorithm for $d = 2$ and $n^{k^{O(1)}}$ time and $n^{O(\log k)}$ space algorithms for $d = 3$. Is there a $n^{O(1)}$ algorithm for $d = 3$?

(3) Dynamic Movement. The problem is to plan the movement of a polygon in d dimensions with bounded velocity modulus between points p_I and p_F, so as to avoid contact with a set O of k polygonal obstacles (defined by a total of n linear constraints) moving with fixed, known velocity. [Reif and Sharir, 85] give the first known investigation of the computational complexity of planning dynamic movement. They show that the problem of planning dynamic movement of a single $(k = 1)$ disk P in $d = 3$ dimensions is polynomial space hard. (This result is somewhat surprising, since P in this case has only 3 degrees of freedom. Our key new idea is to use time to encode a configuration of a polynomial space bounded Turing machine.) Is this problem polynomial space hard for dimension $d = 2$?

Asteroid avoidance problems are a natural subclass of dynamic mover's problems where each obstacle is convex and does not rotate. [Reif and Sharir, 85] give a polynomial time algorithm for dimension $d = 2$ with a bounded number $k = O(1)$ of obstacles and give $2^{n^{O(1)}}$ time and $n^{O(\log n)}$ space algorithms for dimension $d = 3$ with an unbounded number k of obstacles. Is the asteroid avoidance problem polynomial in the case $d = 3$?

1.5 Organization of the Paper

In Section 2.1 we give a precise definition of the generalized mover's problem. In Section 2.2 we define symmetric Turing machines. In Section 2.3 we give the relevant complexity theoretic definitions and results. In Section 2.4, we give our proof that the generalized mover's problem is polynomial space hard.

2. THE GENERALIZED MOVER'S PROBLEM IS PSPACE-HARD

2.1 Definition of the Generalized Mover's Problem

We let a convex polyhedron in three space be specified by a finite set of linear inequalities with rational coefficients. We let a *(rational) polyhedron* be specified by a finite union of such convex polyhedra. Such a polyhedron P can be encoded by some fixed convention as a finite binary string $\langle P \rangle$.

We will formally specify the three dimensional generalized mover's problem (O,P,p_I,p_F) as follows:

(1) the *obstacle set* O consists of a finite set of (rational) polyhedra $O_1,\ldots,O_{n_1}$

(2) the *object to be moved*, P, consists of a finite set of (rational) polyhedra $P_1,\ldots,P_{n_2}$ which are freely linked at distinguished *linkage vertices* $v_1,\ldots,v_{n_3}$

(3) p_I, p_F are distinguished *initial* and *final* rational positions of P.

Hence we may encode (O,P,p_I,p_F) as the string $(\langle O_1 \rangle,\ldots,\langle O_{n_1} \rangle)$ $(\langle P_1 \rangle,\ldots,\langle P_{n_2} \rangle,v_1,\ldots,v_{n_3})$ $(\langle p_I \rangle,\langle p_F \rangle)$. The *size* of (O,P,p_I,p_F) is the length of this encoding.

A *legal position* of P is any position where each polyhedron P_i of P intersects no obstacle of O and furthermore intersect no other polyhedron of P except at its specified linkage vertices. We assume, of course, that p_I and p_F are both legal positions. A *legal movement* of P is a continuous sequence of simultaneous translations and rotations of the polyhedra of P through only legal positions. The *generalized mover's problem* is to determine the existence of a legal movement from p_I to p_F.

It is important to observe that any generalized mover's problem is *reversible* in the sense that if there is a legal movement of P from p_I to p_F, then the movement can always be reversed so as to begin at p_F and end at p_I. This reversibility property imposes a constraint on the class of computation problems which can be simulated by generalized movement problems; in particular the simulated machine must be symmetric in a sense precisely defined below.

2.2 Symmetric Computations

A *symmetric Turing machine* is defined (see also [Lewis and Papadimitriou, 82] for an equivalent definition) as $M = (\Gamma, \Sigma, Q, q_I, q_F, \Delta)$ where

(i) Γ is the *tape alphabet* with distinguished *pad symbol* $\$ \in \Gamma$ and *blank symbol* $\# \in \Gamma$

(ii) $\Sigma \subseteq \Gamma - \{\$, \#\}$ is the *input alphabet*

(iii) Q is the state set with distinguished *initial state* $q_I \in Q$ and *accepting state* $q_F \in Q$

(iv) $\Delta \subseteq (Q \times \Gamma^2 \times \{-1,1\})^2$ is the *transition relation*, where we require that for each transition $((q,L,R,D),(q',L',R',D')) \in \Delta$

(a) $D' = -D$

(b) if $L = \$$, then $D \neq 1$. Alternatively, if $R = \$$, then $D \neq 1$

(c) also $((q',L',R',D'),(q,L,R,D)) \in \Delta$.

We will also be given a *space bound* $s = s(n)$ which is a function of the input length n such that $s(n) \geq n$. M has a single read/write tape with $s+2$ tape cells. This tape has contents $t = t_0 t_1 \ldots t_s t_{s+1}$ where $t_0 = t_{s+1} = \$$ and $t_1, \ldots, t_s \in \Gamma - \{\$\}$.

M has a single read/write tape head which simultaneously scans the tape cell under the current head position, as well as the tape cell immediately to the left or right of the current head position depending on the direction of the next move of the tape head (this convention is used to allow for reversability). Restriction (b) insures M never moves its head off the end of the tape. Restriction (c) implies that the transition relation is a symmetric relation.

More precisely, a *configuration* of M is a tuple $ID = (q,h,t)$ where $q \in Q$ is the current state, $h \in \{1,\ldots,s\}$ is the current position of the tape head, and $t = t_0 t_1 \ldots t_s t_{s+1} \in \$(\Gamma - \{\$\})^s \$$ is the current tape contents. The *next move relation* $\vdash$ is a relation on configurations such that $(q,h,t) \vdash (q',h',t')$ iff there exists a transtion $((q,L,R,D),(q',L',R',D')) \in \Delta$ the new head position is $h' = h + D$, the new tape contents t' are identical to the previous tape contents t except at positions h and $h+D$.

(1) if $D = 1$ then $t_h = L$, $t_{h+1} = R$, $t'_h = L'$, and $t'_{h+1} = R'$

(2) if $D = -1$ then $t_{h-1} = L$, $t_h = R$, $t'_{h-1} = L'$, and $t'_h = R'$.

Given the input string $w = w_1 \ldots w_n \in \Sigma^n$, the *initial configuration* is $ID_I(w) = (q_I, 1, \$w_1 \ldots w_n \#^{s-n}\$)$. We define $ID_F = (q_F, 1, \$\#^s\$)$ to be the accepting configuration of M. Let $\vdash^*$ be the transitive closure of $\vdash$. M *accepts* input w iff $ID_0(w) \vdash^* ID_F$. Let $L(M)$ be the language accepted by M.

2.3 Complexity Definitions

For some space bound $s = s(n) \geq n$ let DSPACE(s), SSPACE(s), NSPACE(s) denote the class of language accepted by deterministic, symmetric, and nondeterministic Turing machines, respectively. [Savitch, 70] shows

PROPOSITION 1. $NSPACE(s) \subseteq DSPACE(s^2)$.

M accepts input w.

We can assume, without loss of generality, that $s = s(n)$ is constructible in deterministic $O(\log n)$ space.

It will be useful to consider the tape alphabet Γ to be the integers $\{1,\ldots,\gamma\}$, where $\gamma = |\Gamma|$.

We begin by defining P, the object which is to be moved. P will contain a sequence $A_0,\ldots,A_{s+1}$ of triangular pyramids of identical size which will be called *arms*. For each $i = 0,\ldots,s+1$ arm A_i has a distinguished *apex vertex* v_i. A_i has an equilateral triangular base with base sides of length $a = 1/(4(\gamma+1))$. Each of the vertices of the base is of length $1/2$ from the apex vertex v_i (see Figure 5). For each $i = 0,\ldots,s$ there is also a straight (one dimensional) link of length 1 from v_i to v_{i+1} which freely links A_i to A_{i+1} (see Figure 6).

It will be useful to define a *cutout polygon* Q consisting of the union of a rectangle and a set of triangles $\{Q_{ij}\}$ of identical size for $i = 0,\ldots,2s+1$ and $j = 1,\ldots,\gamma$. The rectangle is of horizontal length $2s+1$ and vertical height $\varepsilon = a/10$. Each triangle Q_{ij} has a distinguished vertex u_i, connected to two sides of length $1/2+\varepsilon$, and a base side of length $a+\varepsilon$ opposite u_i (see Figure 7). On the upper side of the rectangle is the sequence of vertex $u_0,\ldots,u_{2s+1}$ spaced at distance 1 between each other. For each $i = 0,\ldots,2s+1$ the triangles $Q_{i1},\ldots,Q_{i\gamma}$ each share vertex u_i but are otherwise disjoint, and arranged in cyclic order (as in Figure 8).

Let TUNNEL(Q) be a cylinder with perpendicular cross-section Q. Therefore, the interior of TUNNEL(Q) is formed by sweeping Q in a direction perpendicular to the plane in which Q is contained. We will call the region swept out by triangle $Q_{i,j}$ the $Q_{i,j}$-*slot*.

The basic idea in our construction will be to use the $s+2$ degrees of freedom of P to encode a given configuration of M.

Let $h \in \{1,\ldots,s\}$ be a head position and let $t = t_0 t_1 \ldots t_s t_{s+1} \in \$(\Gamma - \{\$\})^s\$$ be the contents of the tape. We say P *encodes* (h,t) if P is positioned in the interior of TUNNEL(Q) so that for $i = 0,1,\ldots,s+1$ arm A_i is in the Q_{s-h+i,t_i}-slot (see Figure 9). We say P is *properly positioned* if P encodes some (h,t). We shall define obstacles and the initial position in such a way that P is always properly positioned.

Observe that we have defined TUNNEL(Q) so that if P is properly positioned in its interior and P encodes (h,t), then P always encodes (h,t) on any legal movement of P within the interior of TUNNEL(Q) since the arms of P remain in the same slots.

A *segment* of TUNNEL(Q) is a copy of the cylinder TUNNEL(Q) bounded by two planes perpendicular to the cylinder (see Figure 10). We will allow separate segments of TUNNEL(Q) to be merged into a single copy of a TUNNEL(Q) segment. This can be done as in Figure 11, so that if P encodes (h,t) on an entrance, P encodes (h,t) on the exit. Note that of course, P can also move from the exit back to either entrance, without modifying the encoding (h,t). Thus this construction can also be viewed as the branch of a segment of TUNNEL(Q) into two segments of TUNNEL(Q).

Next we require a construction of obstacles which force P to modify its position so as to simulate next moves of the symmetric machine M.

For any $L,R \in \{1,\ldots,\gamma\}$, let Q[L,R] be the polygon derived from Q by deleting all triangles Q_{s,j_0} and Q_{s+1,j_1} for all $j_0 \in \{1,\ldots\gamma\} - \{L\}$ and $j_1 \in \{1,\ldots\gamma\} - \{R\}$ (see Figure 12). Observe that if P is positioned in the interior of TUNNEL(Q[L,R]) and P encodes (h,t), then arm A_h must be in the $Q_{s,L}$-slot and arm A_{h+1} must be in the $Q_{s+1,R}$-slot and hence the encoded tape symbols in the h and h+1 position are $t_h = L$ and $t_{h+1} = R$, respectively.

Let $\hat{Q}$ be the figure derived from Q by adding two semidisks with radius $1/2+\varepsilon$, and with centers at u_s and u_{s+1} (see Figure 13). Note that if P is positioned in the interior of TUNNEL($\hat{Q}$) so that P encodes (h,t) except at t_h and t_{h+1}, then the arms A_h and A_{h+1} are each free to move within the interior region swept out by a semidisk.

Let $\delta \in \Delta$ be a transition, where

$$\delta = ((q,L,R,D),(q',L',R',-D)) \quad .$$

We will define an obstacle B_δ with a connected interior region with distinguished *entrance* and *exit*, and with the property that if P enters the interior of B_δ encoding (h,t), then when P exits B_δ, P encodes (h',t'), where

$$(q,h,t) \vdash (q',h',t') \quad .$$

We first consider the case $D=1$. Then we let B_δ consist of a concatenation of unit length symbols of the following:

(1) TUNNEL(Q)

(2) TUNNEL($Q[L,R]$)

(3) TUNNEL($\hat{Q}$)

(4) TUNNEL($Q[L',R']$)

(5) TUNNEL(Q), which is displaced one unit to the left with respect to segments (1)-(4).

(See Figure 14.)

Suppose P enters B_δ encoding (h,t). Then P can move through TUNNEL($Q[L,R]$) only if $t_h=L$ and $t_{h+1}=R$. After moving through TUNNEL($\hat{Q}$), P encodes (h,t'), where t' is identical to t except t'_h and t'_{h+1} are arbitrary elements of $\{1,\ldots\gamma\}$. However, P can move through TUNNEL(Q[L',R']) only if $t'_h = L'$ and $t'_{h+1} = R'$. Since the last segment of TUNNEL(Q) is displaced one unit to the left, P exits B_δ encoding $(h+1,t')$, where $(q,h,t) \vdash (q',h+1,t')$.

In the case $D=-1$, we take B_δ to be $B_{\delta'}$ with the exit and entrance face reversed, where $\delta' = ((q',L',R',1),(q,L,R,-1))$. (Note that $B_{\delta'}$ is already defined by the above construction for $D=1$.) Since movement of P is always reversible, P enters B_δ encoding (h,t) and exits encoding $(h-1,t')$ iff P enters $B_{\delta'}$ encoding $(h-1,t')$ and exits encoding (h,t) iff $(q',h-1,t') \vdash (q,h,t)$ iff $(q,h,t) \vdash (q',h-1,t')$, since $\vdash$ is symmetric.

We now have defined all the elementary building blocks required to simulate a computation of M. We will construct a copy C_q of a TUNNEL(Q) segment for each state $q \in Q$. C_q will make a series of branches so as to lead to the entrance of each B_δ such that $\delta \in \Delta$ is a transition from state q. Also C_q will make a series of branches in the opposite direction, so as to lead to the exit of each $B_{\delta'}$ such that $\delta' \in \Delta$ is a transition to state q. Note that the construction is of polynomial size and can easily be done by a O(log n) space deterministic Turing machine.

For the proof of our construction, it will be useful to extend our definition of encoding so that if P is located in the interior of C_q encoding (h,t), we also then say that P *encodes configuration* $ID = (q,h,t)$.

Given input $w = w_1 \ldots w_n \in W^n$, we define the *initial position* p_I to be a rational position of P encoding the initial configuration $ID_0(w) = (q_I, 1, \$w_1 \ldots w_n \#^{s-n}\$)$.

The *final position* p_F is defined to be a rational position of P encoding the accepting configuration $ID_F = (q_F, 1, \$\#^s\$)$.

LEMMA. P *has a legal movement from* $p_I(w)$ *to a position encoding configuration* ID *iff* $ID_0(w) \vdash^* ID$.

Proof. $ID_0(w) \vdash^* ID$ iff $\exists$ a sequence of configurations $ID_0(w) = ID_0, ID_1, \ldots, ID_k = ID$ where $ID_0 \vdash ID_1, \ldots, ID_{k-1} \vdash ID_k$ iff $\exists$ a sequence of transitions $\delta_1, \ldots, \delta_k \in \Delta$ where $ID_i = \delta_i(ID_{i-1})$ for $i = 1, \ldots, k$. We now claim that this holds iff P has a legal movement from $p_I(w)$ through $B_{\delta_1}, \ldots, B_{\delta_k}$ (in this order) to a position p_k encoding $ID_k = ID$. In the case $k = 0$, the claim obviously holds since $p_I(w)$ encodes $ID_0(e)$. Suppose the claim holds for all $k' < k$. Then P has a legal movement from $p_I(w)$ through $B_{\delta_1}, \ldots, B_{\delta_{k-1}}$ to position p_{k-1} encoding ID_{k-1} iff $ID_0(w) \vdash^* ID_{k-1}$. But our above construction of B_{δ_k} insures that there exists a legal movement of P from p_{k-1} through B_{δ_k} to a position encoding ID_k iff $ID_k = \delta_k(ID_{k-1})$. Hence the claim holds. □

The Lemma then implies: P has a legal movement from initial position $p_I(w)$ to final position p_F iff $ID_0(w) \vdash^* ID_F$, where ID_F is the accepting configuration. This completes the proof of our theorem. □

ACKNOWLEDGEMENTS

Thanks to Martin Cherez, Christoph Freytag, Debby Joseph, and Paul Spirakis for helpful comments on this paper.

REFERENCES

M. Brady, J.M. Hollerbach, T.L. Johnson, T. Lozano-Pérez, and M.T. Mason (eds.), *Robot Motion: Planning and Control*, MIT Press, 1983.

M. Ben-Or, D. Kozen, and J.H. Reif, "Complexity of elementary algebra and geometry", *16th Symposium on Theory of Computing*, 1984; also to appear in *J. Computer and System Sciences*, 1985.

B. Chazelle, "A theorem on polygon cutting with applications", *Proc. 23rd IEEE Symp. on Foundations of Computer Science*, Chicago, IL, 339-349, 1982.

G.E. Collins, "Quantifier elimination for real closed fields by cylindric algebraic decomposition," *Proc. 2nd GI Conference on Automata Theory and Formal Languages*, Springer-Verlag LNCS 35, Berlin, 134-183, 1975.

P.G. Comba, "A procedure for detecting intersections of three-dimensional objects," *J. ACM* 15, 3, 354-366, July 1968.

M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freedman and Co., San Francisco, 1979.

J.E. Hopcroft, J.T. Schwartz and M. Sharir, "On the complexity of motion planning for multiply independent objects: PSPACE hardness of the warehouseman's problem," TR-103, Courant Institute Mathematics, Feb., 1984.

J.E. Hopcroft, D.A. Joseph, and S.H. Whitesides, "On the movement of robot arms in 2-dimensional bounded regions," *Proc. 23rd IEEE Symp. of Foundations of Computer Science*, Chicago, IL, 280-289, 1982.

H.R. Lewis and C.H. Papadimitriou, "Symmetric space bounded computation," *Theor. Comput. Sci.* 19, 161-187, 1982.

T. Lozano-Pérez and M. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *CACM* 22, 560-570, 1979.

G. Miller and J.H. Reif, "Robotic movement planning in the presence of friction is undecidable," to appear, 1985.

N.J. Nilsson, "A mobile automation: An application of artificial intelligence techniques," *Proceedings IJCAI-69*, 509-520, 1969.

C. O'Dunlaing, M. Sharir, and C.K. Yap, "Retraction: A new approach to motion planning," *Proc. 15th ACM Symp. on the Theory of Computing*, Boston, MA, 207-220, 1983.

R. Paul, "Modelling trajectory calculation and servoing of a computer controlled arm," Ph.D. thesis, Stanford University, Nov. 1972.

V.T. Rajan and J.T. Schwartz, work in progress, 1985.

J.H. Reif, "Complexity of the mover's problem," *Proc. 20th IEEE Symp. on Foundations of Computer Science*, San Juan, Puerto Rico, 421-427, 1979.

J.H. Reif and J. Storer, "Shortest paths in Euclidean space with polyhedral obstacles," Center for Research in Computing Technology, Harvard University, TR-05-85, May 1985.

J.H. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," Center for Research in Computing Technology, Harvard University, TR-06-85, May 1985.

W.J. Savitch, "Relationships between nondeterministic and deterministic tape complexities," *J. Computer Sci.* 4, 177-192, 1970.

J.T. Schwartz and M. Sharir, "On the piano movers' problem: I. The special case of a rigid polygonal body moving amidst polygonal barriers," *Comm. Pure Appl. Math.* Vol. XXXVI, 345-398, 1983.

J.T. Schwartz and M. Sharir, "On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds," *Adv. Appl. Math.* 4, 298-351, 1983.

J.T. Schwartz and M. Sharir, "On the piano movers' problem: III. Coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers," The Int'l. J. of Robotics Research, vol. 2, no. 3, Fall 1983, pp.46-75.

M. Sharir and A. Schorr, "On shortest paths in polyhedral spaces," *Proc. 16th ACM Symp. on the Theory of Computing*, Washington, DC, 144-153, 1984.

[Lewis and Papadimitriou , 82] show

PROPOSITION 2. $\mathrm{DSPACE}(s) \subseteq \mathrm{SSPACE}(s) \subseteq \mathrm{NSPACE}(s)$.

Let

$$\mathrm{PSPACE} = \bigcup_{c \geq 1} \mathrm{DSPACE}(n^c) \quad .$$

The above imply

PROPOSITION 3. $\mathrm{PSPACE} = \bigcup_{c \geq 1} \mathrm{SSPACE}(n^c)$.

A *log-space reduction* from a language $L' \subseteq \Sigma^*$ to a language L is a mapping f computable by a $O(\log n)$ space bounded deterministic Turing machine such that for each input $w \in \Sigma^*$, $w \in L'$ iff $f(w) \in L$. In this case, we say L' is *log-space reducible* to L. Note that any log-space reduction requires only time bound $2^{O(\log n)} = n^{O(1)}$.

Given a language class $\mathcal{L}$, a language L is $\mathcal{L}$*-hard* if each language $L' \in \mathcal{L}$ is log-space reducible to L.

2.4 The Simu· ion of a Symmetric Turing Machine

We now prove:

THEOREM. *The generalized mover's problem is* PSPACE-*hard.*

Proof. Let $M = (\Gamma, \Sigma, Q, q_I, q_F, \Delta)$ be a symmetric Turing machine with polynomial space bound $s(n) = n^c$ for some constant $c \geq 1$. We will construct a log-space reduction from $L(M)$ to the generalized mover's problem. In particular, given an input $w = w_1 \ldots w_n \in \Sigma^n$, we must construct in $O(\log n)$ space a mover's problem $f(w) = (O, P, p_I, p_F)$ such that P has a legal movement from p_I to p_F iff

P. Spirakis and C. Yap, "Strong NP-hardness of moving many discs," to appear, *Information Processing Letters*, Aug. 1985.

S. Udupa, "Collision detection and avoidance in computer controlled manipulators," Cal. Inst. Tech., Ph.D. thesis, 1977.

C. Widdoes, "A heuristic collision avoider for the Stanford robot arm," Stanford CS Memo 227, June 1974.

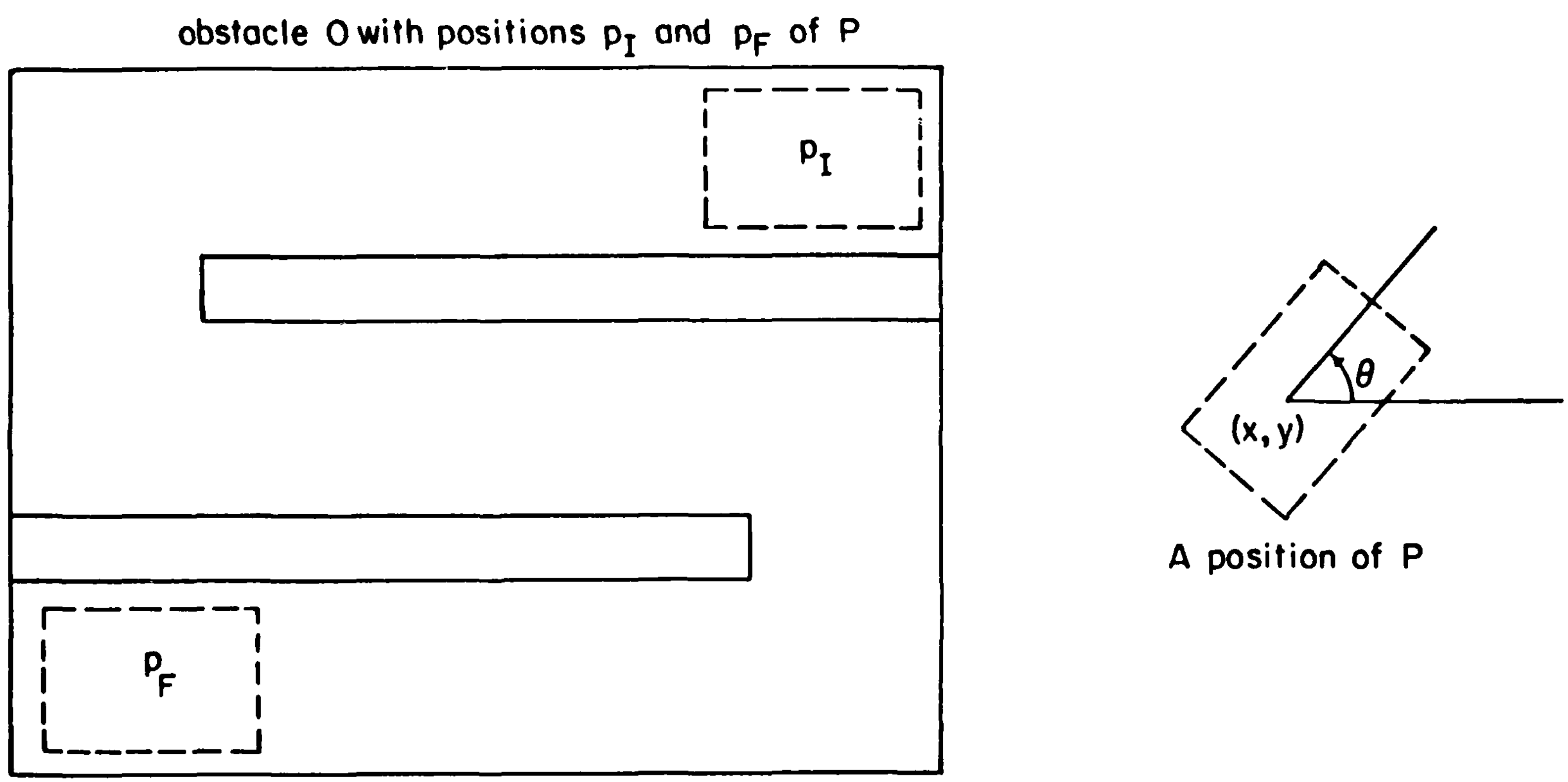


Figure 1. A 2-D Mover's Problem: Can rectangle P be moved from p_I to p_F without contacting an obstacle in O?

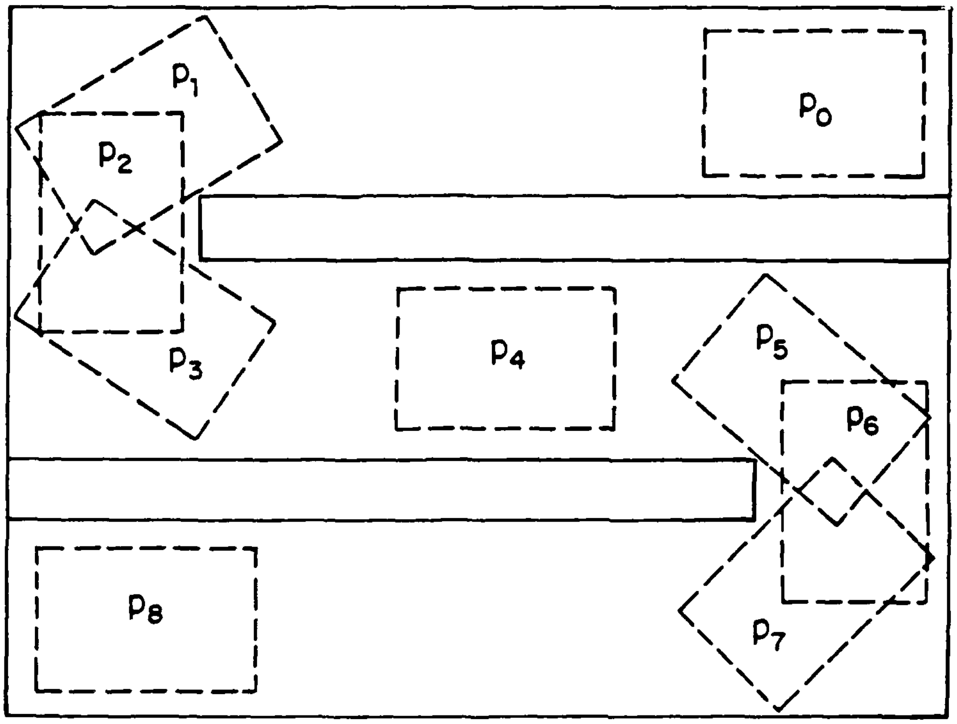


Figure 2. A solution to the 2-D Mover's Problem of Figure 1. P may be moved through positions $p_I = p_0, p_1, \ldots, p_8 = p_F$.

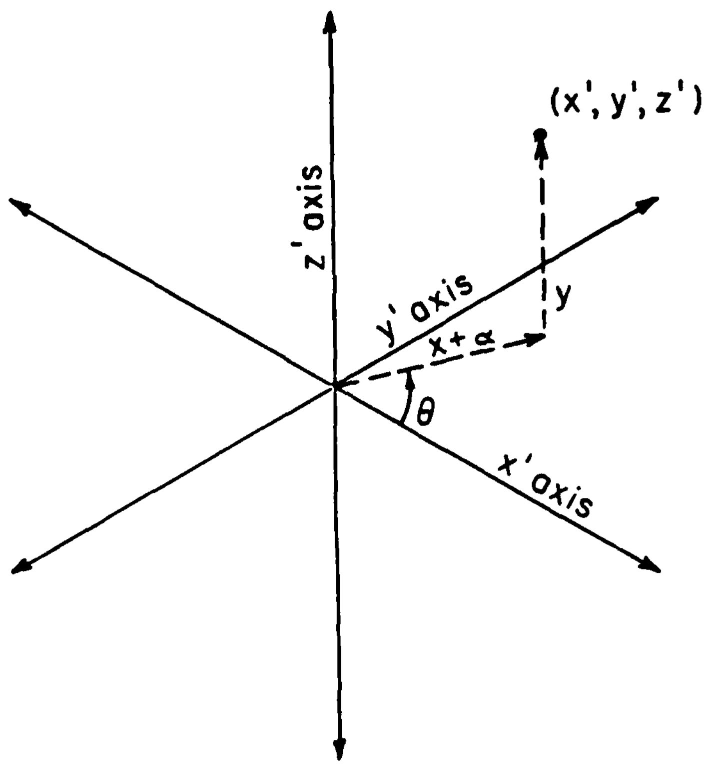


Figure 3. The mapping $f(x,y,\theta) = (x',y',z')$.

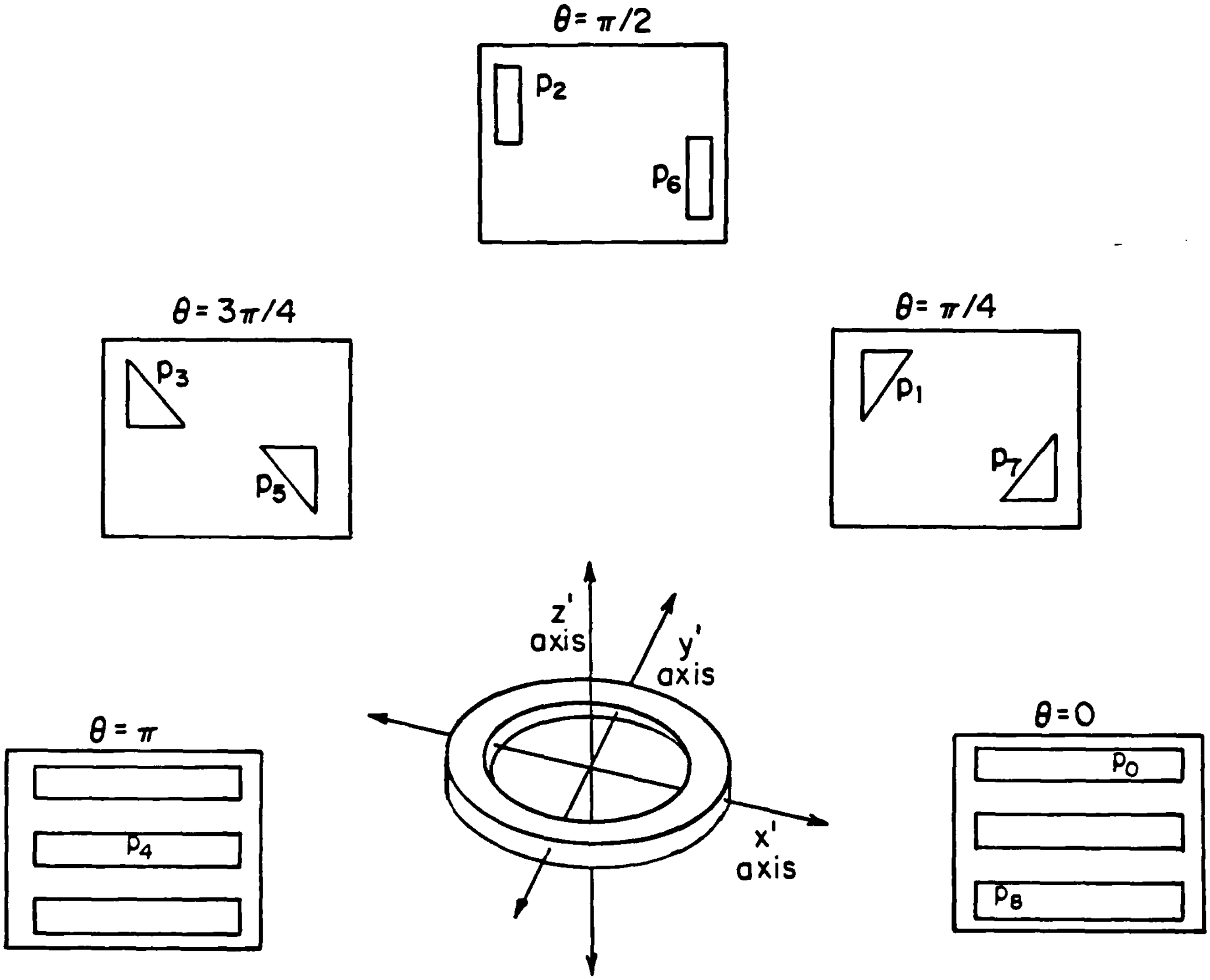


Figure 4. Transformed mover's problem from Figure 1. The obstacles of the transformed problem define a torus with cross-sections illustrated for $\theta = 0, \pi/4, \pi/2, 3\pi/4, \pi$. P may be moved through positions $p_I = p_0, p_1, \ldots, p_8 = p_F$ as in Figure 2.

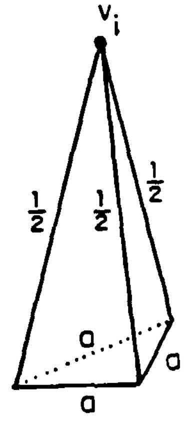


Figure 5. An arm A_i.

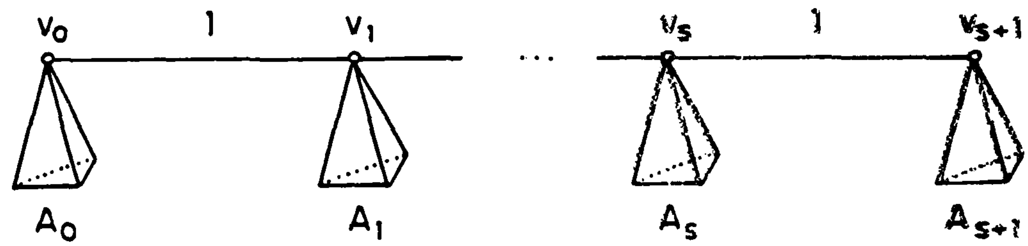


Figure 6. P, the object to be moved.

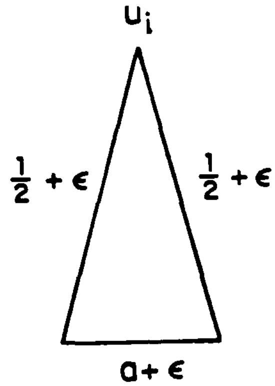


Figure 7. A triangle Q_{ij}.

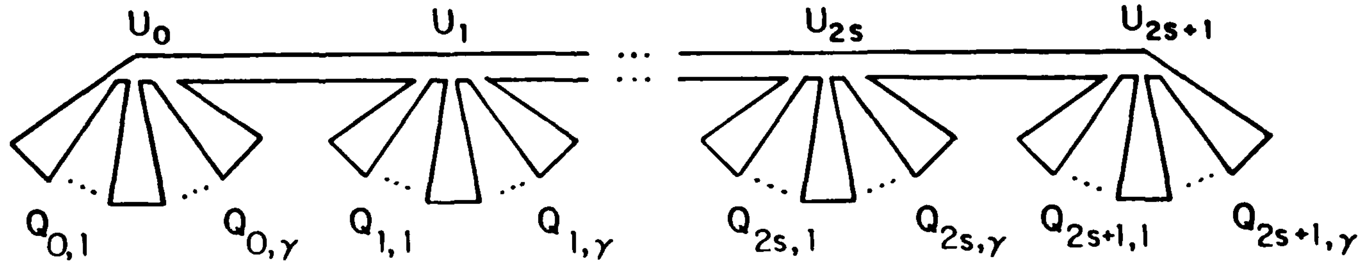


Figure 8. The polygon Q.

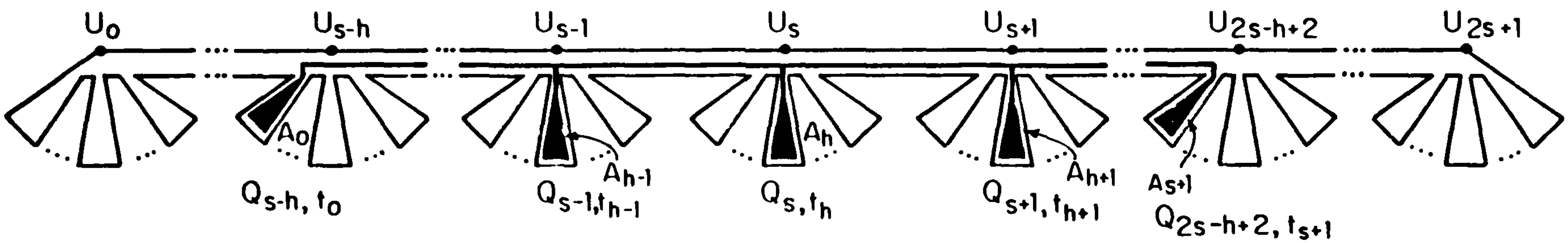


Figure 9. A position of P encoding (h,t) where $t = t_0t_1 \ldots t_{s+1}$.

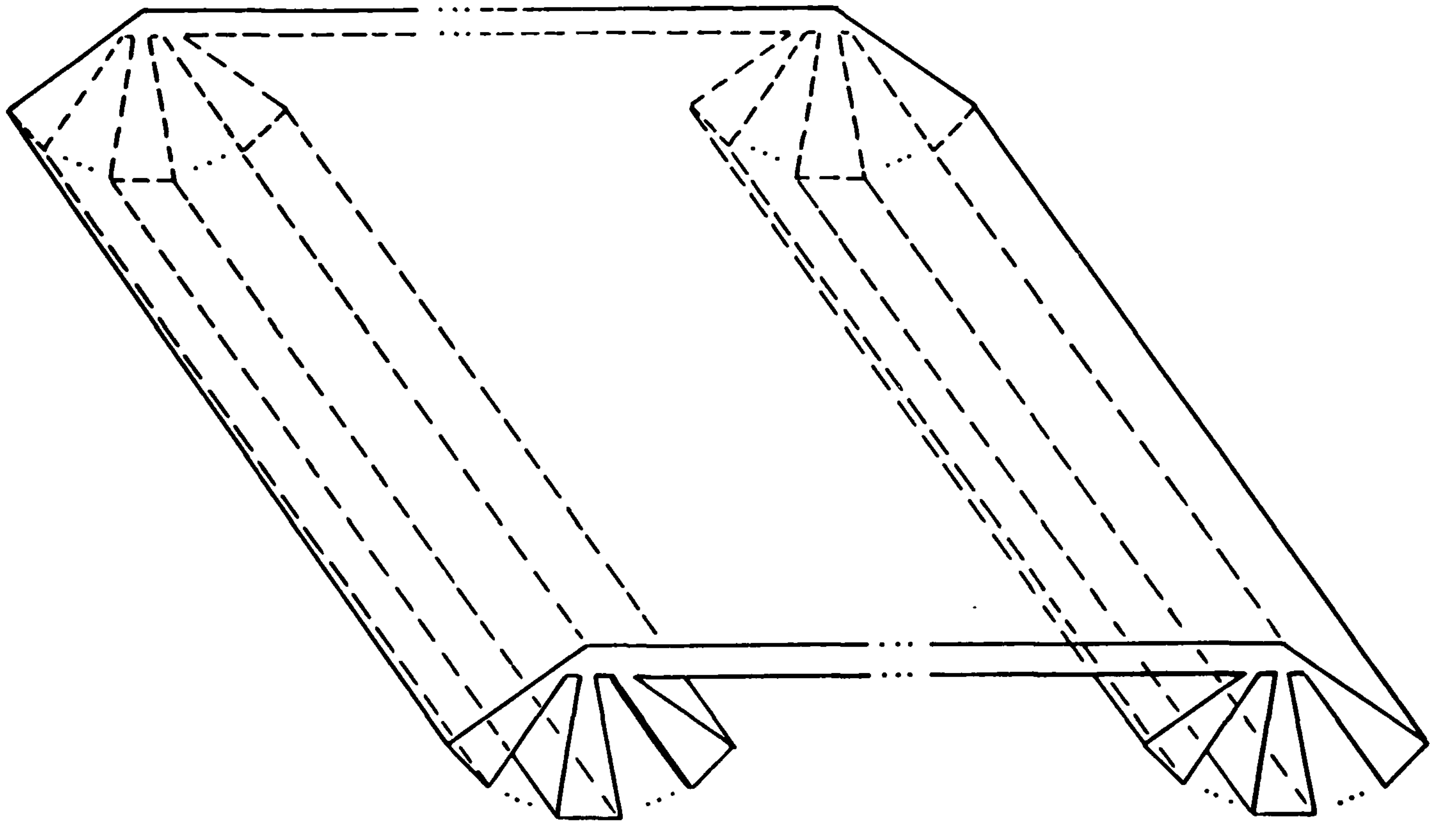


Figure 10. A segment of TUNNEL(O).

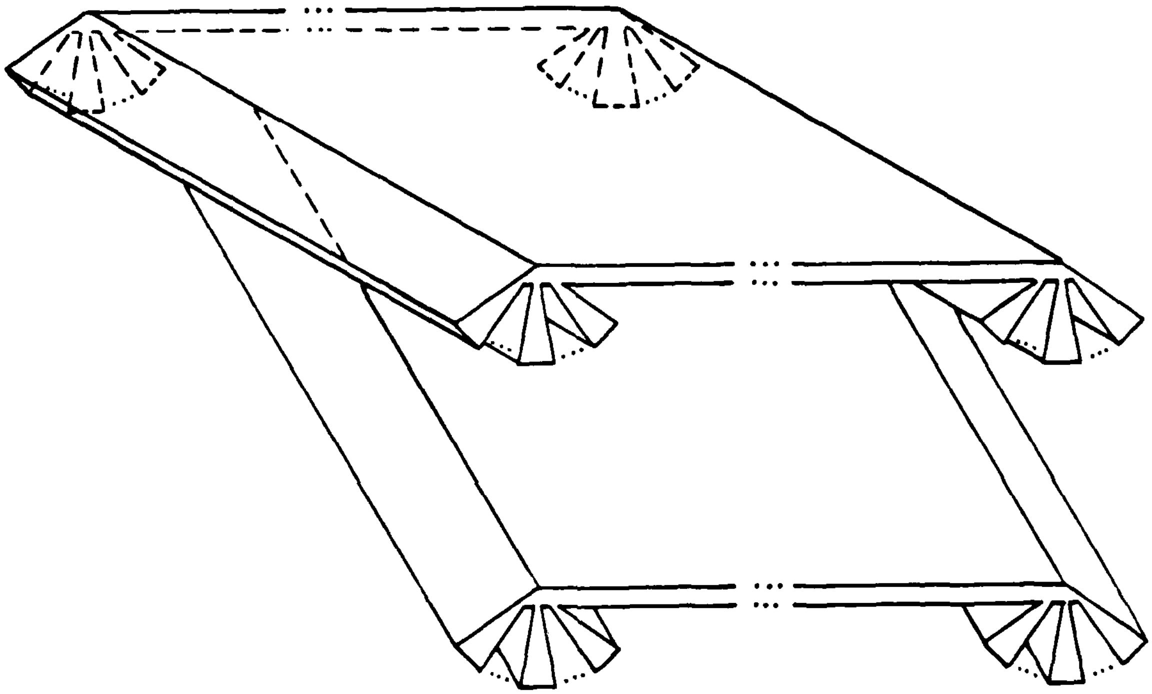


Figure 11. The merge of two segments of TUNNEL(Q).

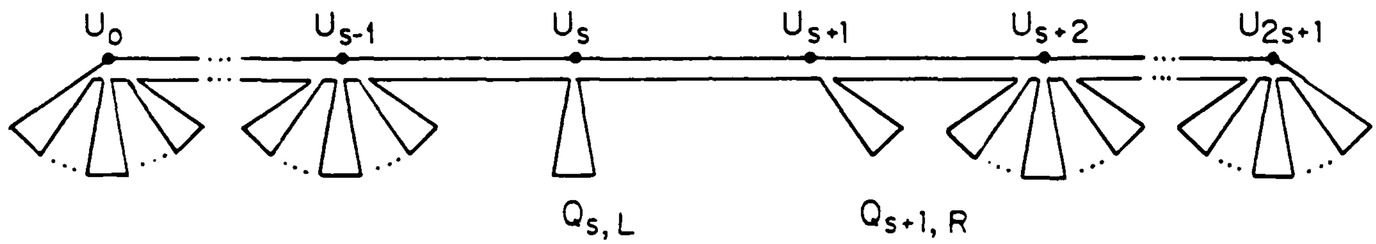


Figure 12. The polygon $Q[L,R]$.

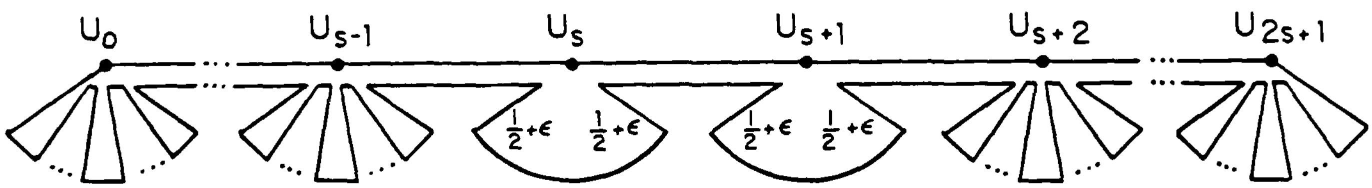


Figure 13. The polygon $\hat{Q}$.

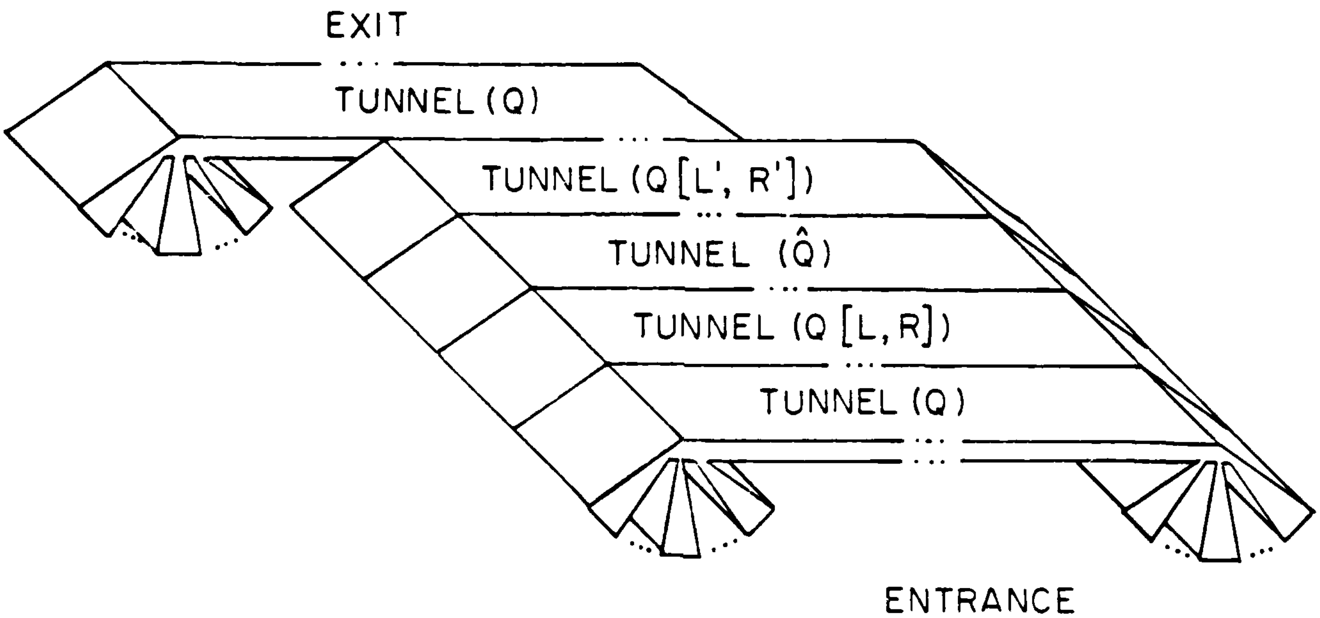


Figure 14. The obstacle $B_{\hat{Q}}$.

END

FILMED

1-86

DTIC