rmation in predicate calculus.
Michie, D.). Edinburgh:

uestion-answering systems.

essing conceptual information.
Department, Uppsala

ctual. *Philosophy of Science*,

ro-Planner Reference Manual.
nbridge, Mass: MIT.

12

A Heuristic Solution to the Tangram Puzzle

E. S. Deutsch and K. C. Hayes Jr.
Computer Science Center
University of Maryland

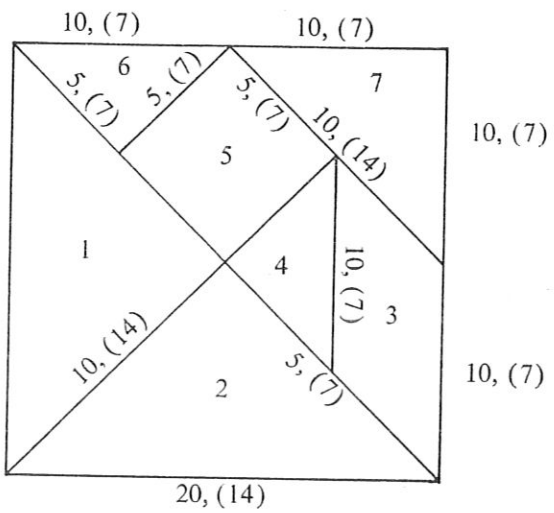
Abstract

A heuristic program leading to the solution of tangram puzzles is described. The program extracts puzzle pieces using a set of rules which search for piece-defining edges. The rules decrease in their rigor, and hence in their reliability, in the sense that the edge requirements become more lax. Such edges include those constructed during the solution process. Composites of puzzles are also formed and are treated like puzzle pieces. The solution procedure is such that the most reliable rules are applied recursively as often as possible. It is only when the solution process comes to a halt that the lower reliability rules are applied in order for the process to continue. Sometimes it is necessary to commence with one of the weaker rules after which a return to the more reliable rules is made.

1. INTRODUCTION

The tangram problem is essentially a puzzle problem but its solution has very little in common with the solution to the familiar jigsaw puzzle (Freeman and Garder 1964). For whereas the jigsaw is composed of a host of pieces of many different shapes, the tangram consists of seven pieces only, all of which have a very simple shape. Furthermore, the way in which the jigsaw puzzle pieces are combined is unique; with the tangram there is a great variety of ways in which the seven pieces can be combined so as to form different shapes.

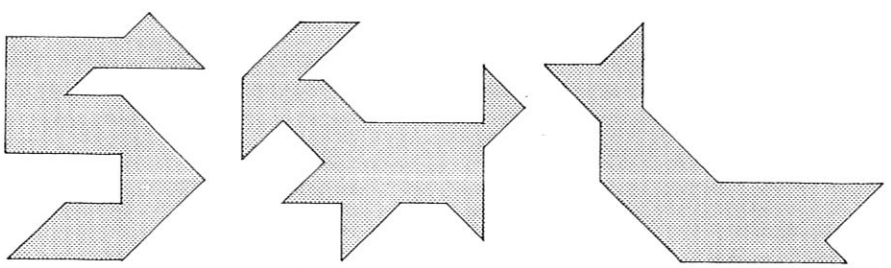
The tangram consists of the following seven pieces: two large right triangles, two small right triangles, one medium-size right triangle, a square and a rhomboid. They can all be combined to form the puzzle shown in figure 1(a). Two points of interest should be noted. Firstly, all the pieces have edges inclined to one another at an angle which is a multiple of 45°. Secondly, it should be observed that the pieces are either three or four sided. The first



(a)

Figure 1(a)

property facilitates the display of a tangram puzzle on a rectangular matrix of points for analysis purposes. This does not apply to the tangram globally, as some edges may be inclined at angles other than multiples of 45°. The second property reduces the number of common edges two or more pieces may share.



(b)

Figure 1(b)

Given a tangram shape, examples of which are given in figure 1(b), the problem consists of assembling the seven tangram pieces in such a manner so that the final assembly corresponds to the given shape. This report discusses a possible computer solution to this problem. It should be stated that the solution suggested is not a generalized one in the sense that it does not purport to solve every tangram puzzle available. Our source of tangram

puzzles is Read's book *Tang*
puzzles employing two tangr
holes. These types of puzzle
was considered better to co

2. GENERAL APPRO

At least two approaches con
In the first instance one coul
computer program would at
variety of ways until, by t
combination rules would cl
approach would definitely ha
A principal objection to thi
involved.

On the other hand, a solut
heuristic programming. Wit
which would first perform a
tions on the given shape. The
in such a way that the contri
an individual piece, or a set
the remaining pieces. Ideally
facilities by means of whic
information gleaned from
program should incorporate
afresh, using another set of t
approach to be followed her

The solution to the tangra
the need for the ranking of
tangram puzzles showed tha
the puzzle than did others
invaluable later on during
tests had little further effect
solved without resorting to
initiate the solution process.
heuristics yielded nothing at
later section where this poin
establishing an ordered list c

Now, it could be argued tl
It should, however, be realiz
degree of interdependence b
taken up, even provisionally
as yet unplaced, pieces. For
the solution process that the
must then be exercised by th

puzzles is Read's book *Tangrams* (Read 1965), and it contains, for example, puzzles employing two tangram sets (fourteen pieces), and puzzles containing holes. These types of puzzles are more complicated in their structure and it was considered better to concentrate on a solution to simpler shapes first.

2. GENERAL APPROACH

At least two approaches come to mind with reference to this kind of problem. In the first instance one could resort to the use of combinatorics, whereby the computer program would attempt to combine the seven tangram pieces in a variety of ways until, by trial and error, a solution was reached. Some combination rules would clearly be incorporated in this program and the approach would definitely have a characteristic combinatorics flavor about it. A principal objection to this approach is the large amount of computation involved.

On the other hand, a solution could be attempted from the point of view of heuristic programming. With this approach one would develop a program which would first perform a series of tests and tentative partitioning operations on the given shape. These would then be evaluated somehow, or ordered, in such a way that the contribution of each test would be used to decide how an individual piece, or a set of pieces, should be placed in conjunction with the remaining pieces. Ideally, the program should incorporate back-tracking facilities by means of which the analysis procedure can resort to some information gleaned from a previously discarded test. Furthermore, the program should incorporate a facility whereby the analysis can be started afresh, using another set of tests, if no solution is obtained so far. This is the approach to be followed here.

The solution to the tangram problem to be described below hinges upon the need for the ranking of tests and rules. Examination of a number of tangram puzzles showed that some tests contributed more to the solution of the puzzle than did others. However, the latters' contribution becomes invaluable later on during the solution process, once the earlier valuable tests had little further effect. At the same time, some puzzles could not be solved without resorting to the less powerful tests first, merely in order to initiate the solution process. In these cases the usually more powerful tests or heuristics yielded nothing at all. Specific examples will be dealt with in a later section where this point will be amplified; however, the desirability of establishing an ordered list of heuristics should be clear.

Now, it could be argued that all the tests should be performed in parallel. It should, however, be realized that as the solution proceeds, there is a high degree of interdependence between tangram pieces that have already been taken up, even provisionally, to form a partial solution and the remaining, as yet unplaced, pieces. For example, it may be decided at a certain stage in the solution process that the square be placed in a particular position. Care must then be exercised by the program as to whether a similar effect would

not be realizable had the two smaller triangles been placed in that position instead. The square piece would then be used for a later insertion. Alternatively, the square piece may no longer be available at this stage, in which case the solution method must incorporate the possibility of substituting the two small triangular pieces for the square piece. Since there is considerable interdependence between the partial solution and the as-yet-unsolved remainder of the puzzle, the ranking of tests and back-tracking are quite necessary.

The following strategy was adopted for the mechanized solution of the tangram problem. Various *global* operations are to be applied to the entire given puzzle first. The purpose of these operations would be to split the puzzle into, hopefully, easier subpuzzles. In addition, these operations would yield some tentative guidelines as to the way in which some of the individual tangram pieces should be placed in the puzzle. A solution, entire or partial, is then attempted using the heuristic considered most powerful. The heuristic is applied to the subpuzzles too, if any, and will rely in part upon results obtained by the preliminary operations. In most instances only a partial solution will be obtained at this stage; the preliminary set of operations and the first heuristic are re-applied. As soon as the solution's progress is halted the remainder of the unsolved puzzle is analyzed by the heuristic next in rank. Should a further partial solution be derived, the remainder of the puzzle is re-investigated by the first heuristic. The heuristic ranking third is applied should the second heuristic yield nothing. Each time a heuristic places a puzzle piece, even tentatively, an immediate return to the most powerful heuristic is made. Should the puzzle be unsolved by the time all the heuristics have been applied, a fresh attempt is made: this time, however, the second-ranking heuristic is applied first, and so on. A return to the first heuristic is made as soon as a partial solution is available.

In summary, the reasoning behind the structure of the program is the following. Clearly the most reliable heuristics, in the sense that they are deemed to provide a correct solution more often than not, are the ones that should be used as frequently as possible. Experience with those used here has shown that when their criteria are met, the resulting placement of a puzzle piece is very reliable. The criteria of these high-ranking heuristics are rather stringent, and should the solution process terminate unsuccessfully on using them, the next most reliable one should be used if only to continue the solution process. The requirements of the next most reliable heuristics will be less stringent, and their contribution is only admitted as a bridging step to the next step, which is a return to the use of the first set of heuristics. In this way, the ensuing portion of the solution is as reliable as possible. Figure 2 is a generalized block diagram of the program's structure. Details of the individual blocks are discussed in the next sections.

A similar argument holds for the insertion of entry points E_1, E_2 in figure 2. It may well happen that a complete solution will not be reached after the

application of all the heurist
delete the contributions of eac
this time starting at E_2. The
used first, might assist in findi

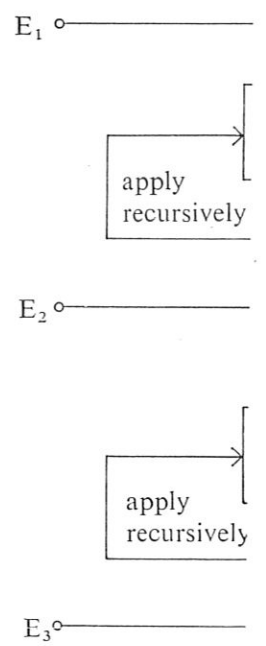


Figure 2

Finally, it is pointed out that
local. This approach differs fr
himself to three-dimensional
Given a set of lines describin
used to establish which block
kinds of local properties car
tangram puzzles since only th
properties are to be used.

3. SOME GEOMETR

Before describing the details
types of tangram puzzles to b

application of all the heuristics. In that case it may be advantageous to delete the contributions of each of the heuristics and start the problem afresh this time starting at E_2. The reduced requirements of the second heuristic, used first, might assist in finding the correct solution the next time around.

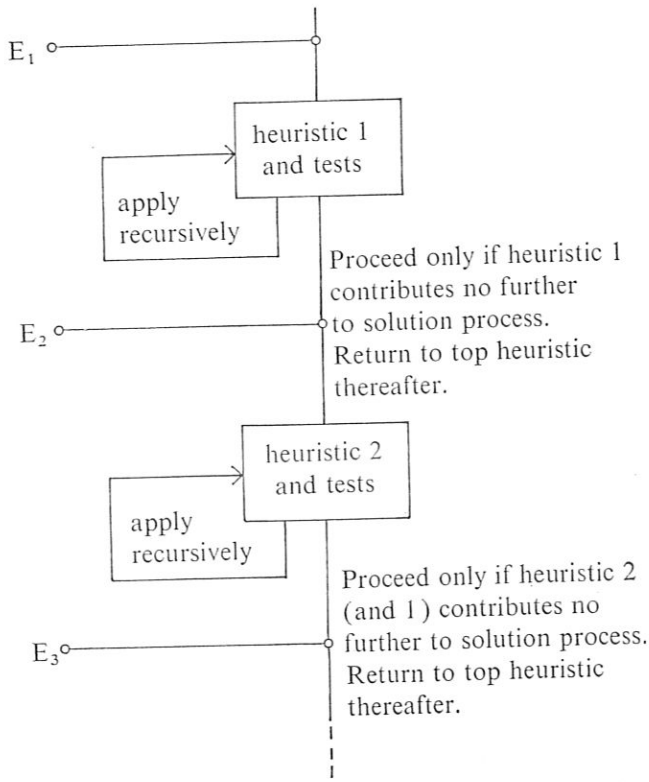


Figure 2

Finally, it is pointed out that the tests to be described are global rather than local. This approach differs from that of Guzman (1968), who, in addressing himself to three-dimensional scene analysis, used local edge properties. Given a set of lines describing the scene, the extracted local properties were used to establish which blocks, for example, partially occluded others. These kinds of local properties cannot be derived directly when one deals with tangram puzzles since only the puzzle's periphery is given. Thus global edge properties are to be used.

3. SOME GEOMETRIC CONSIDERATIONS

Before describing the details of the program, a few words concerning the types of tangram puzzles to be considered here are necessary. It was realized

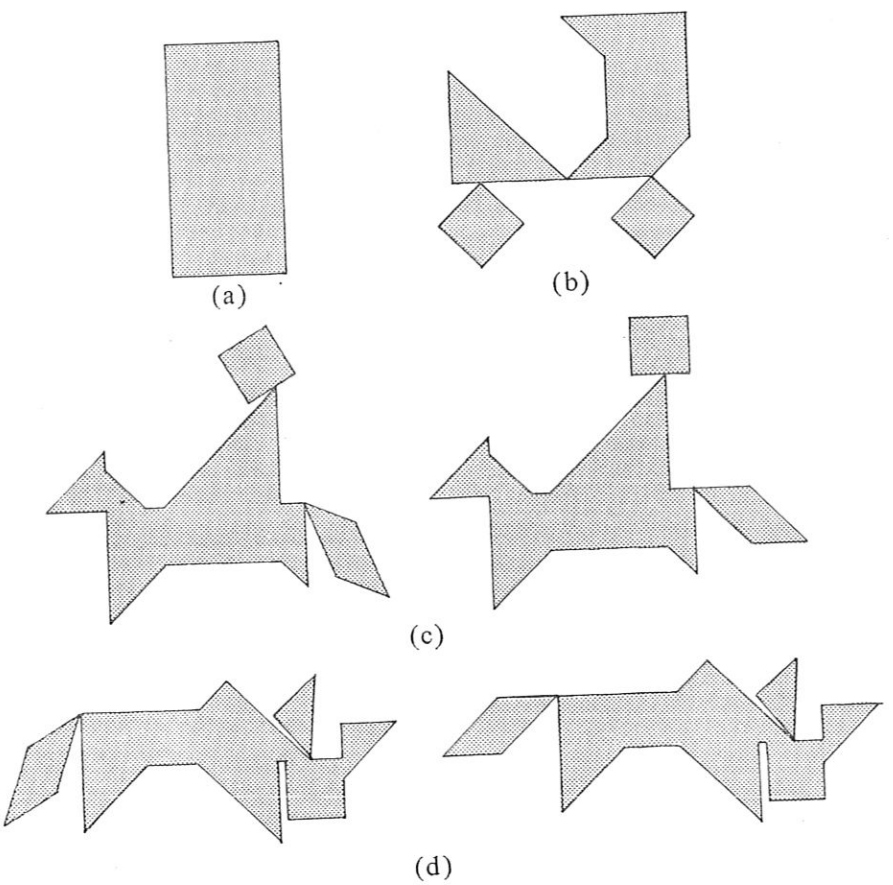


Figure 3

quite early on that the num
contains considerable inforn
location of the individual pu
grams shown in figure 3. T
figure 3(a), provides very l
tangram in figure 3(b) can, h
corners indicate how some c
later much information is a
outline since it suggests the
Rather than also attempt th
fewer corners, it was decide
eleven corners or more, in t
would at least be obtainable

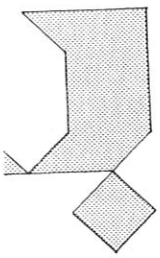
A further restriction on
requirement that all the puz
puzzles containing loops and
locate the internal puzzle-ec
necessity of having to handle
avoiding. Luckily the numbe

Of course it is possible to
puzzles, employing two se
tangrams. While the single
may be equally applicable t
here.

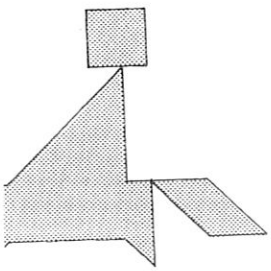
Some puzzles may contain
a multiple of 45°. In some c
form. This presents some p
array is used. Where the p
changing the puzzle's overall
was discarded. Such tangrar
caused contact between nor
loop. With the tangram in f
tangram of figure 3(d) it wa
nature of the puzzle. Inclin
3(e) for an example. Genera
well formed unless all of its
multiple of 45°.

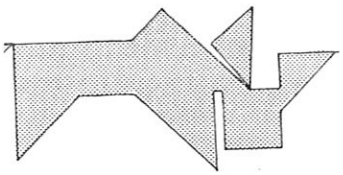
Figure 1(a) also gives th
pieces in terms of inter-pictu
with digitized line drawings
points and diagonal picture
ignored. This is not the case
numbers shown along the ed
picture point distances) at th

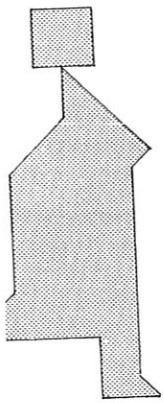
P



(b)







quite early on that the number of corners or vertices of a tangram shape contains considerable information about its possible structure and thus the location of the individual puzzle pieces within it. Consider the first two tangrams shown in figure 3. The outline of the four-cornered convex shape, figure 3(a), provides very little information concerning its structure. The tangram in figure 3(b) can, however, be solved more easily since the numerous corners indicate how some of the pieces might be placed. As will be shown later much information is also contained in the concave portions of the outline since it suggests the existence of contact between two puzzle pieces. Rather than also attempt the solution of the 'tougher' puzzles, containing fewer corners, it was decided to opt for the solution to puzzles containing eleven corners or more, in the hope that a solution for the 'easier' puzzles would at least be obtainable.

A further restriction on the type of tangrams to be considered is the requirement that all the puzzles be filled in. This requirement excludes all puzzles containing loops and holes. The complexity involved in first having to locate the internal puzzle-edges circumscribing the holes, followed by the necessity of having to handle two sets of edges at a time was considered worth avoiding. Luckily the number of such puzzles is very limited.

Of course it is possible to consider puzzles made of fourteen pieces. Such puzzles, employing two sets of seven puzzle pieces, are called double tangrams. While the single-tangram techniques, to be described below, may be equally applicable to double tangrams we will not consider them here.

Some puzzles may contain edges which are not inclined at an angle that is a multiple of 45°. In some cases the entire tangram is given in an inclined form. This presents some problems, due to sampling, when a rectangular array is used. Where the portion of a tangram could be rotated without changing the puzzle's overall structure this was done. Otherwise, the tangram was discarded. Such tangrams would be those in which reorientation either caused contact between non-touching edges or caused the formation of a loop. With the tangram in figure 3(c) reorientation was possible; with the tangram of figure 3(d) it was not possible without materially changing the nature of the puzzle. Inclined tangrams were similarly rotated; see figure 3(e) for an example. Generally, then, a tangram puzzle was not considered well formed unless all of its edges were inclined at an angle whose value is a multiple of 45°.

Figure 1(a) also gives the dimensions used for the individual tangram pieces in terms of inter-picture point distances. In many problems associated with digitized line drawings, the fact that distances between axial picture points and diagonal picture points differ by a factor of $\sqrt{2}$ can be safely ignored. This is not the case here, because true lengths are important. The numbers shown along the edges in figure 1(a) give the length of that edge (in picture point distances) at the current orientation of that edge. The number

in brackets signifies the length of that edge when its inclination is changed from axial to diagonal or vice versa.

One of the problems associated with finding a solution to a puzzle is the fact that the eye may be deceived as to the actual length of an edge comprising the puzzle shape, and short of using a ruler, the length of the edge must be guessed. The guessed length could have been used as input to the program, but this might present a host of side problems that would only stifle the heuristics. Accordingly, lengths were read off the puzzle shapes directly.

4. TANGRAM ANALYSIS

4.1 Subpuzzle determination and extension line construction

The puzzle is input to the machine in terms of its vertices' numbers (corners) and edge lengths. The practice was to start with the leftmost, topmost vertex on the puzzle and proceed in a clockwise direction. The puzzle shape was traced out on an (x, y) array.

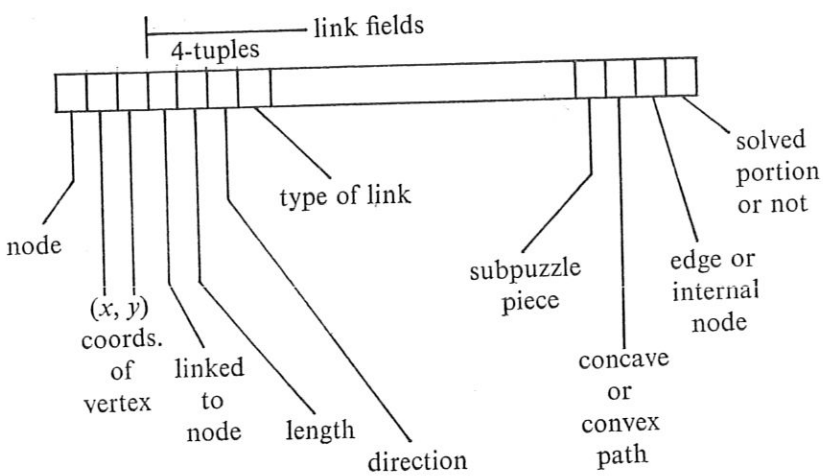


Figure 4

Associated with each vertex is a one-dimensional array the contents of which will be explained as the analysis procedure is developed. The first few items in the array are the vertex's number, and its (x, y) coordinates in the array (see figure 4). Each vertex is connected to two other vertices through two edges of known length and orientation. Each vertex connected to the vertex in question is represented by a 4-tuple within the vertex array. The numbers of the connected vertices, the lengths and direction of the connecting edges are entered into the 4-tuples. The fourth position within each tuple indicates some arc information and is explained later.

The program first attempts to separate the puzzle into two or more subpuzzles. This implies that the separation procedure must ensure that the

subpuzzles do not share a cor
portions of the puzzle meet
such severance points are loc
on the (x, y) plane. A retur
starting point, implies a poir
edges traced during the first
The remaining portion of the
is continued until a return to
figure 3(c) comprises three
number associated with each
can also be separated along a
into the subpuzzle area an
vertex. Figure 5 shows som

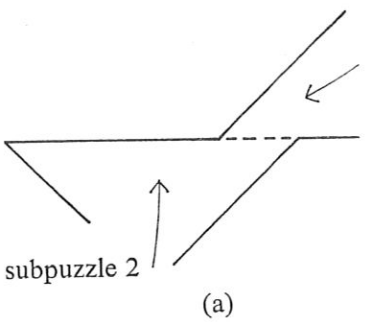


Figure 5

Note that in the case of figu
relative to the subpuzzle nu
Another item on the vertex a
which each vertex belongs.

Puzzle vertices fall into tw
a convex edge path and thos
Determination of the type o
change in the edge at each v
important since the contribu
they are thus treated differen
strongly indicates a possible
identified with the location o
or more puzzle pieces meet.
protruding corners contain
information is available at t
latter points are extended
revealing the possible locati

subpuzzles do not share a common piece at the position of separation. Where portions of the puzzle meet at a point, the subdivision is clear. The way such severance points are located is by tracing along the edges of the tangram on the (x, y) plane. A return to an already traced point, other than to the starting point, implies a point contact. The tangram portion defined by the edges traced during the first and second encounter constitutes a subpuzzle. The remaining portion of the puzzle is another subpuzzle and the procedure is continued until a return to the starting point is made. The shape shown in figure 3(c) comprises three subpuzzles. Subpuzzles are numbered and the number associated with each vertex is entered in the vertex array. Subpuzzles can also be separated along an internal edge. This edge is formed by extending into the subpuzzle area an edge whose end is very close to a neighboring vertex. Figure 5 shows some examples of subpuzzles formed in this way.

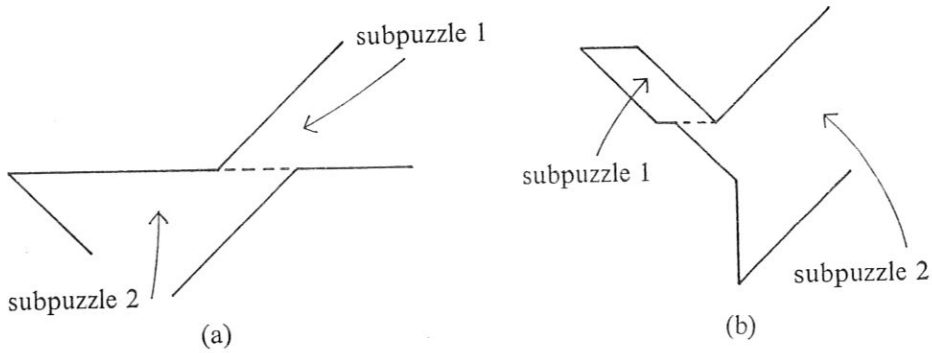


Figure 5

Note that in the case of figure 5(a), the right hand vertex has no meaning relative to the subpuzzle numbered 1. It is erased from the vertex array. Another item on the vertex array is therefore the subpuzzle piece number to which each vertex belongs.

Puzzle vertices fall into two categories: those connecting edges which form a convex edge path and those connecting edges which form a concave path. Determination of the type of each vertex is made by measuring the angular change in the edge at each vertex. The distinction between these two types is important since the contribution of each type to the solution is different and they are thus treated differently. The corner formed by a convex-type vertex strongly indicates a possible puzzle piece fit, whereas a concave type vertex is identified with the location of a line (inside the puzzle area) along which two or more puzzle pieces meet. Consider figure 6(a): it will be observed that the protruding corners contain considerable piece-fit information. No such information is available at the recessed vertices. However, if the edges at the latter points are extended back into the puzzle area a set of guide lines revealing the possible location of further pieces as well as piece-adjacency

information is obtained. The lines so obtained will be referred to as extension lines (see figure 6(b)). Subsequently, when attempts at fitting puzzle pieces are made, only convex paths need be considered, seeing that the concave paths assist in the construction stage only.

Figure 6(b) shows all the possible extension lines. As these are created new

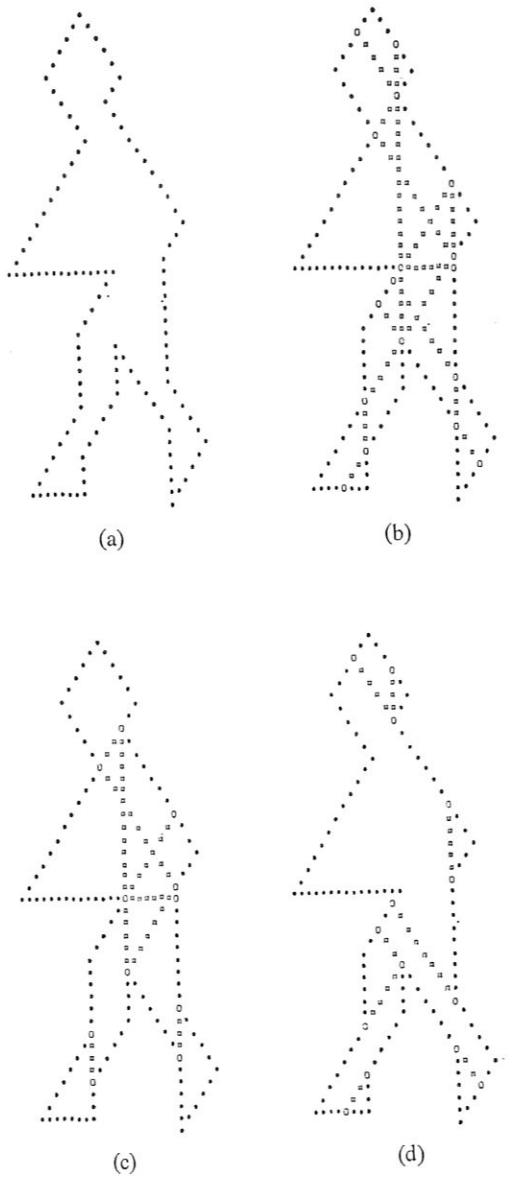


Figure 6

vertices (or nodes) are form
(The words 'nodes' and 've
Not all of the extension line
misleading guide lines and h
is too short to support a
puzzle pieces as in figure 1,
direction whose length was
lines in the diagonal directi
be removed. Extension lines
times.

Extension lines were also
admissible length was remo
puzzle. The distance criteria
extension lines. The reason
piece could possibly lie en
parallel lines. However, exte
retained even if they were tc
boundary of a puzzle piece.
6(c); those eliminated are s

During the extension-line
(or vertices) are generated.
intersection of two or more
intersections of an extension
the vertex array and an entry
new nodes are generated tl
nodes in the array must be
nections. For example, the
vertices must be reflected
connected directly. Further
be entered or updated. The
contents of the 4-tuples w
tuplet indicates whether the
Eventually some internal r
pieces are removed from
severed now becomes an ed

The preliminary tests o
extension lines and the upda
begin. We start by describi
application is discussed late

4.2 Puzzle pieces: direct-m

The program first attempts
rather than by extension li
lines. The seven puzzle piece

vertices (or nodes) are formed both on the edges and within the puzzle area. (The words 'nodes' and 'vertices' will be used interchangeably henceforth.) Not all of the extension lines prove to be useful; some of them can in fact be misleading guide lines and have to be removed. Extension lines whose length is too short to support a puzzle piece are removed. Using the length of puzzle pieces as in figure 1, it was decided that extension lines in the axial direction whose length was less than seven were to be removed. Extension lines in the diagonal direction whose length was less than five were also to be removed. Extension lines terminating at an edge vertex were retained at all times.

Extension lines were also tested for parallelism in that an extension line of admissible length was removed if it was too close to a parallel edge of the puzzle. The distance criteria were the same as above for axial and diagonal extension lines. The reason for the removal of these lines is that no puzzle piece could possibly lie entirely within the region bounded by these two parallel lines. However, extension lines running parallel to one another were retained even if they were too close, since either one of them could form the boundary of a puzzle piece. The extension lines retained are shown in figure 6(c); those eliminated are shown in figure 6(d).

During the extension-line generation procedure, two new types of nodes (or vertices) are generated. One type is formed inside the puzzle area at the intersection of two or more extension lines, and the other, at the edges, at the intersections of an extension line with an edge. These new nodes are added to the vertex array and an entry as to the type is also made (see figure 4). As the new nodes are generated the information related to individual vertices or nodes in the array must be updated in order to indicate the new interconnections. For example, the formation of a node on an edge between two vertices must be reflected in the array showing that they are no longer connected directly. Furthermore, new extension line interconnections must be entered or updated. The updating is made by altering or adding to the contents of the 4-tuples with each vertex array. The fourth entry in the tuplet indicates whether the connection is via an edge or an extension line. Eventually some internal nodes will become edge nodes as solved puzzle pieces are removed from the puzzle. The edge along which a piece was severed now becomes an edge.

The preliminary tests over – subpuzzle separation, the generation of extension lines and the updating of the vertex list – the solution procedure can begin. We start by describing the various rules that are used; their order of application is discussed later.

4.2 Puzzle pieces: direct-match rule

The program first attempts to locate puzzle pieces *fully described by edges*, rather than by extension lines or by combinations of edges and extension lines. The seven puzzle pieces are stored in the machine as are their reflections

and rotations (in steps of 45°) if different from the corresponding puzzle piece initially stored. The (x, y) plane on which the tangram was originally traced is examined in conjunction with the vertex arrays. The distance and orientation of neighboring vertices are examined at each vertex, in order to determine whether the vertex pattern at the vertex in question matches that of a stored puzzle piece. Once a match is found the corresponding portion is severed from the puzzle and an indication to the effect that the matched puzzle piece is no longer available is made.

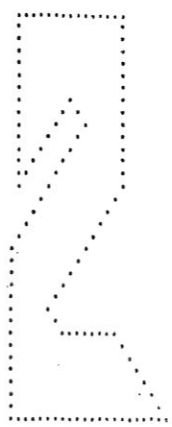
The extraction of a puzzle piece by direct matching occurs in the neighborhood where two subpuzzles meet; the matched piece usually forms one of the subpuzzles (see figures 3(c,d,e)).

4.3 Puzzle pieces: $2\frac{1}{2}$-$3\frac{1}{2}$ edge-match rule

The method of determining the position of puzzle pieces by the rule just described, while being very reliable, cannot be used so frequently in practice for it is not often that a puzzle piece fully bounded by edges is found. Thus some relaxation in the rigor of fit has to be made. Another matching heuristic allows a puzzle piece to be located if most of its periphery is described by edges and only a small portion is described by the constructed extension lines. Stated precisely, the $2\frac{1}{2}$–$3\frac{1}{2}$ edge-match rule requires that in the case of triangular shapes *two complete sides must be defined by edges and the remaining side can be defined by a combination of collinear edges and extension lines. Moreover, the combination must include at least one portion of edge. For four sided puzzle pieces the rule requires that the additonal side also be fully described by an edge.*

This heuristic was found to be extremely powerful in locating puzzle pieces. It should be noted that the rule only requires that the third or fourth side of the puzzle piece be defined by a combination of edges and extension lines, that is, their order of appearance as well as their number is unimportant. The tangram of figure 7(a) shows how a partial solution is obtained for a puzzle by the application of this rule. The three largest triangular pieces are extracted in this way; the smaller remaining rhomboid is matched subsequently with a puzzle piece (the significance of the portion labelled B will be explained later). No priority as to the fit containing the greater amount of peripheral edge in the remaining side is given. An argument for having such a priority is illustrated using figure 7(b). Observe that there are two ways of placing the large triangular piece in the upper portion of the puzzle. It could occupy position ABC or BCD, but not both. A priority based on the amount of edge in the remaining side would have the large triangle placed at top left rather than in the top right portion. In this case, however, a complete solution is obtained eventually either way, but in general, a wrong choice may result in failure and a new fit will be sought.

Despite the seemingly loose definition of puzzle pieces by this rule, wrong extraction of pieces occurs rarely. In fact, some puzzles may be completely

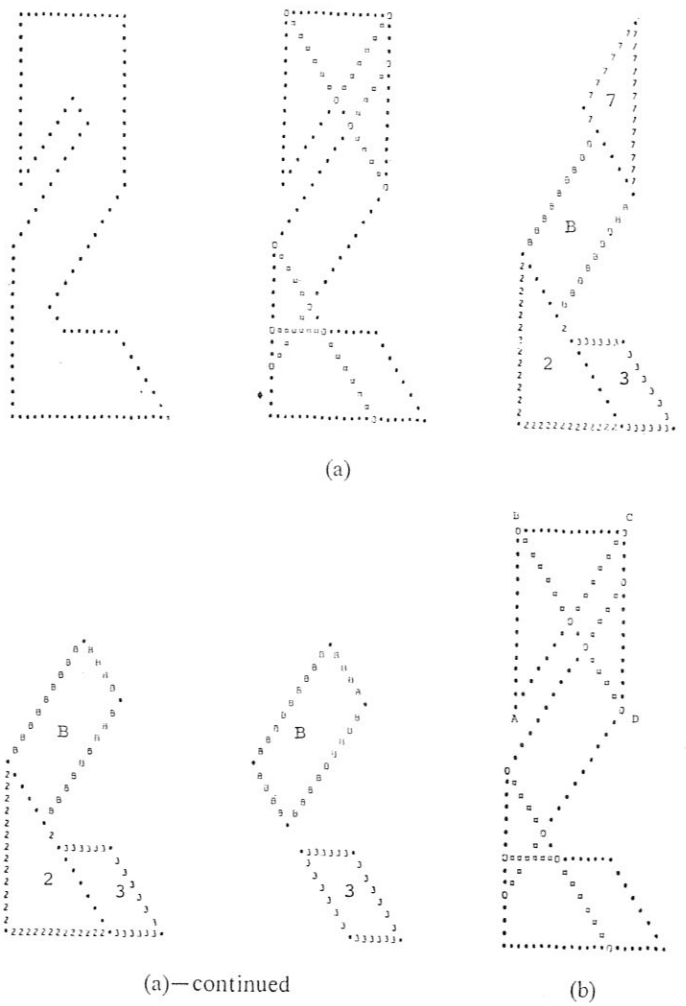


(a)—co

Figure 7

solved by applying the direc
rule just described. Such a p
figure 8. Note that in figur
puzzle pieces are identified.

Two updating procedure
(x, y) plane. First, the line
of the puzzle is treated as an e
Secondly, any vertex along
becomes an edge vertex. Thi
puzzle is removed.



(a)

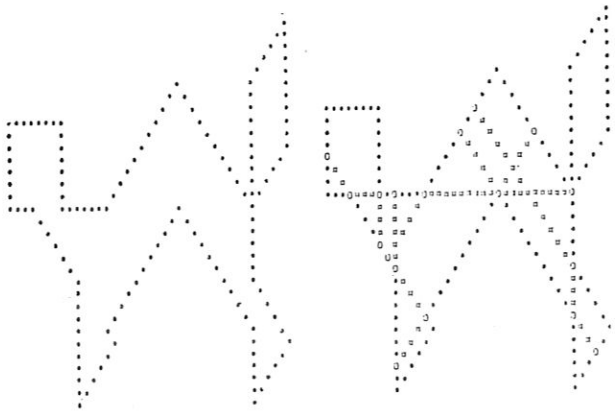
(a)—continued

(b)

Figure 7

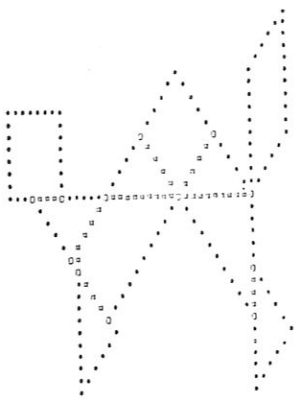
solved by applying the direct-match rule and a repetitive application of the rule just described. Such a puzzle is shown together with its full solution in figure 8. Note that in figure 8(e) the numbers refer to the order in which puzzle pieces are identified. This is true for all subsequent figures.

Two updating procedures now occur within the vertex array and the (x, y) plane. First, the line shared by the removed piece and the remainder of the puzzle is treated as an edge henceforth, since it now lies on the periphery. Secondly, any vertex along this line considered hitherto as internal now becomes an edge vertex. This updating takes place each time a portion of the puzzle is removed.



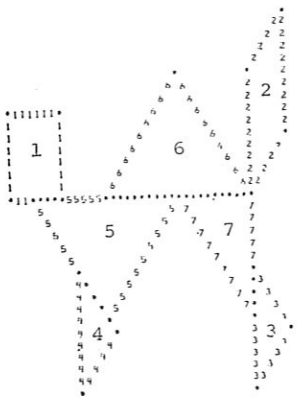
(a)

(b)

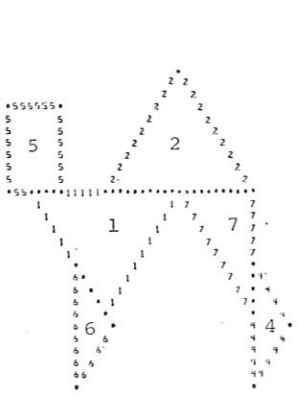


(c)

(d)

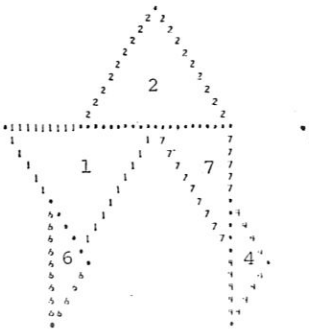


(e)



(f)

Figure 8



(g)

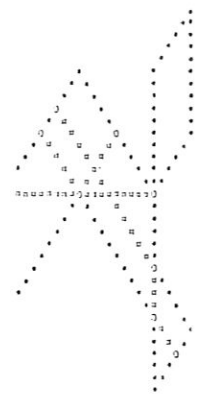
(j)

Figure 8 contd.

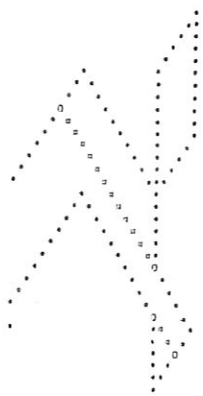
4.4 Composites: direct-matc

It is clear that the rigor of tl
extension lines could be rela
requiring there to be only th
etc, be they edges or extens
ranking of useful heuristics w
further progress towards a s
Before these type of rules are
This involves the extraction
rather than the extraction of
of puzzle pieces are called c

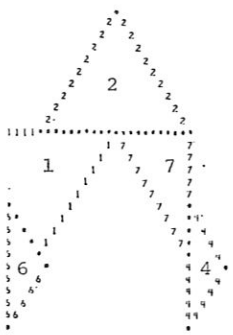
Consider the two puzzles
sented by dots. In particular
the rule described in the
ABCDEFGA denotes the un



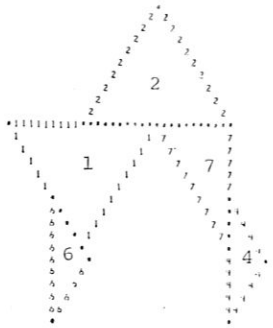
(b)



(d)

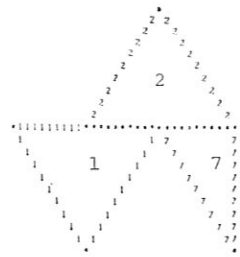


(f)

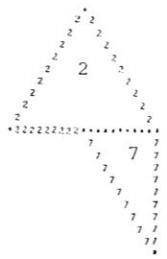


(g)

(h)

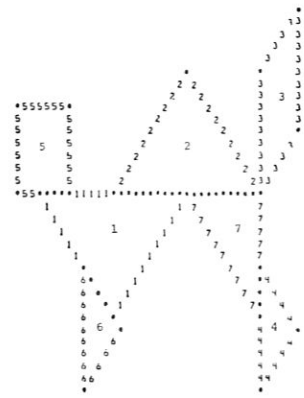


(i)



(j)

(k)



(l)

Figure 8 contd.

4.4 Composites: direct-match rule

It is clear that the rigor of the matching requirements in terms of edges and extension lines could be relaxed further. As a matter of fact a matching rule requiring there to be only three out of the four edges of a four sided piece, etc, be they edges or extension lines, could be used. Their position in the ranking of useful heuristics would be low; nevertheless, in the absence of any further progress towards a solution such rules may have to be resorted to. Before these type of rules are discussed a more powerful heuristic is presented. This involves the extraction of a specific group of puzzle pieces all at once rather than the extraction of single puzzle pieces one at a time. Such groups of puzzle pieces are called composites.

Consider the two puzzles shown in figure 9. The extension lines are represented by dots. In particular consider what happens after the application of the rule described in the previous section. In the case of figure 9(a), ABCDEFGA denotes the unsolved part of the puzzle. In figure 9(b) the

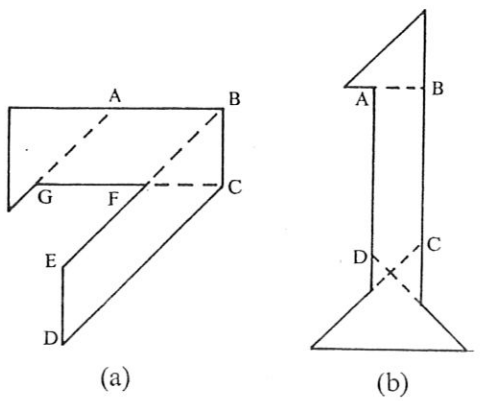


Figure 9

unsolved portion of the puzzle is ABCDA. In both cases there is no further indication as to how the puzzle pieces are distributed. Clearly the as-yet-undiscovered pieces combine in some way to fit the unsolved portion exactly. It appears generally, and it is a contention of this approach, that the unsolved tangram can be partitioned into composite shapes which assume a relatively simple outline. In principle, one form a composite shape could take is the shape outlined by ABCDEFGA. If one pursues this idea a little further one could conceive a variety of types of composite forms of varying degrees of complexity. The formation of these could be a lengthy task and would hinder the generality of the approach. Instead, simpler composites can be used such as ABCFGA and CDEFC, both of which combine to form the remainder of the puzzle. The portion labelled B in figure 7(a) is an example of a composite. In the case of figure 9(b) the puzzle contains the simple composite ABCDA. In obtaining the composites, the problem is not only the generation of simple composite shapes but also how few composites will be sufficient.

A first restriction on the formation of composites is that they must be either three or four sided. The reason for this is that composite shapes will be treated as if they were puzzle pieces and the same edge rules will be applied to them. Also, this restriction limits the complexity of the composites. It will be observed that a given composite can be formed by combining different pieces in more than one way. Consider the four-sided composite shown in figure 10(a). It can be formed in three different ways using a different set of puzzle pieces in each case (see figure 10(b) through (d)). (There are actually six different ways of constructing the composite since there are two small triangular shapes. However, the two pieces are interchangeable.) Note that in figure 10(e) the puzzle pieces of figure 10(d) have just been organized in a different way and therefore do not constitute a distinct composite structure.

Some of the composites can assume the shape of the actual puzzle pieces. The large triangular piece, for example, can be formed using other puzzle

pieces. Puzzle pieces are als
solution process they are treat
sible. An identified puzzle pie
individual piece is no longer a

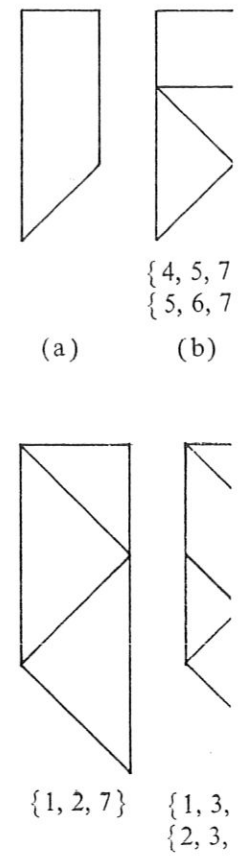
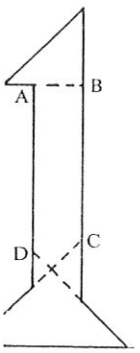


Figure 10

The composite shown in fi
sites, also exists in a similar
larger composite is formed. O
of composites, its correspond
the list. Composites are stor
are the seven puzzle pieces.

Associated with each com
form the composite. As soon



(b)

ɔoth cases there is no further
stributed. Clearly the as-yet-
the unsolved portion exactly.
s approach, that the unsolved
pes which assume a relatively
osite shape could take is the
this idea a little further one
forms of varying degrees of
engthy task and would hinder
composites can be used such
to form the remainder of the
an example of a composite.
he simple composite ABCDA.
not only the generation of
ıposites will be sufficient.
ıposites is that they must be
that composite shapes will be
ıe edge rules will be applied to
y of the composites. It will be
by combining different pieces
d composite shown in figure
using a different set of puzzle
(d)). (There are actually six
ɜince there are two small tri-
ıterchangeable.) Note that in
have just been organized in a
. distinct composite structure.
ɔe of the actual puzzle pieces.
ɔe formed using other puzzle

pieces. Puzzle pieces are also considered as composites, however, in the solution process they are treated as individual puzzle pieces as often as possible. An identified puzzle piece will only be treated as a composite once the individual piece is no longer available.

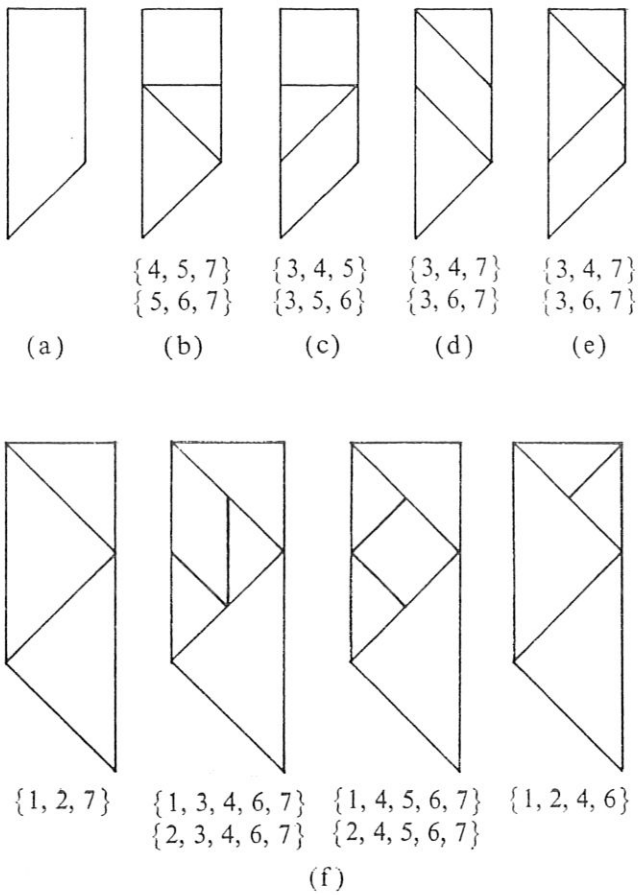


Figure 10

The composite shown in figure 10, as is the case with many other composites, also exists in a similar but enlarged form. Figure 10(f) shows how the larger composite is formed. Once a composite is formed and inserted in the list of composites, its corresponding smaller or larger composite is also added to the list. Composites are stored in the computer in exactly the same way as are the seven puzzle pieces.

Associated with each composite is the list of sets of puzzle pieces which form the composite. As soon as a composite is identified an abeyance list of

these sets is formed. Any set including a puzzle piece no longer available is excluded from the abeyance list. An abeyance list containing no sets whatsoever is an indication that the selected composite cannot be considered, and the solution process continues without this composite having been extracted. As the solution proceeds and more puzzle pieces are placed the abeyance lists corresponding to the identified composites are curtailed as these pieces become unavailable. A complete solution using the abeyance list is obtained at the end after all the remaining unplaced puzzle pieces are distributed so that each composite abeyance list contains one set which is satisfied. The character associated with each composite indicates the composite's identity in the printout. When any of the twenty-one composites are involved in a solution, they are identified by the appropriate character and the appropriate set of puzzle pieces from the abeyance list. Note that a composite in the shape of a puzzle piece will appear with its appropriate composite character.

The fact that any given puzzle can contain only three composites at most is very useful since this can be used as a check to see if an extracted composite is admissible. Furthermore, the fact that there is a maximum of only three composite pieces per puzzle speeds up the processing of the abeyance lists.

The composite piece of figure 9(b) is fully described by edges once the two smaller triangles are identified by the $2\frac{1}{2}$–$3\frac{1}{2}$ rule. The first rule associated with composites, the direct-match rule, *permits the extraction of those composites which are fully defined by edges.* The two composites of figure 9(a) will not be extracted by the rule just stated since each composite is partially defined by an extension line. Composites such as these will be recognized by a later rule.

4.5 Composites: $2\frac{1}{2}$-$3\frac{1}{2}$ edge-match rule

Just as with the individual puzzle pieces, the $2\frac{1}{2}$–$3\frac{1}{2}$ *rule is applied to composites.* It may be argued that there is no need to have separate direct-matching rules and $2\frac{1}{2}$–$3\frac{1}{2}$ edge rules for both the puzzle pieces and the composites, since the program tests them in an identical manner. Eventually, when the heuristics are organized, this may well turn out to be the case. For the moment, however, rules remain separate for each group of shapes. This only means that a particular rule will operate either on the set of puzzle pieces or on the twenty-one composite pieces but the same portion of the program is used in each case.

4.6 Composites: 2-3 edge-match rule

This rule requires still less stringent matching criteria for the extraction of composites. All a potential composite piece must satisfy is that *two of its sides be defined by edges and the remaining side be defined by an extension line. In the case of a four sided composite, three sides must be defined by edges* (see figure 9(a)).

There is one additional requirement, however, which also applies to the rule described in the previous section. Ordinarily, when the composite is

fully described by edges, there
to be extracted. However, whe
may be extracted at any given
embedded in the left vertical li

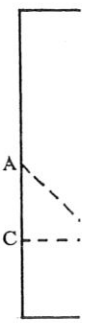


Figure 11

Composite E would have exte
site F would have its fourth s
the above rule require that t
case the rule would extract c

4.7 Puzzle pieces 2-3 edge-m

We now return to further r
The rule under this heading
pieces. This rule never fails
but it can introduce errors wl
The rule is designed to isol
piece can be isolated using
solution obtained using the
example introduces the need
triangular piece in figure 12

fully described by edges, there is no doubt as to the identity of the composite to be extracted. However, when this rule is relaxed, more than one composite may be extracted at any given time. Consider figure 11: two composites are embedded in the left vertical limb of the puzzle, composite E and composite F.

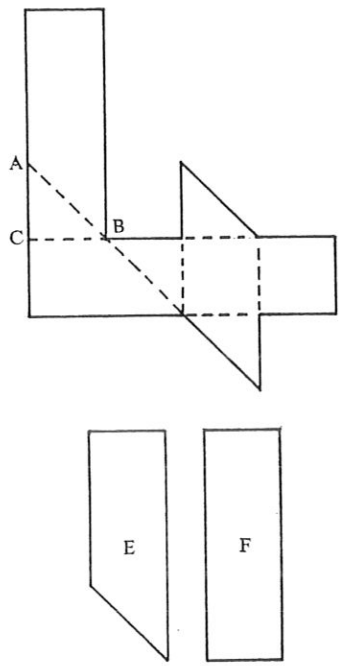


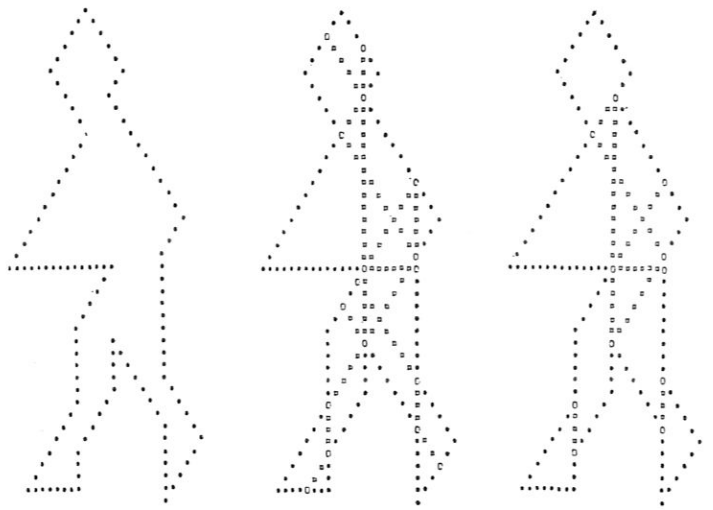
Figure 11

Composite E would have extension line AB as its fourth side, whereas composite F would have its fourth side formed by extension line BC. Both this and the above rule require that the *smallest composite piece be extracted.* In this case the rule would extract composite E.

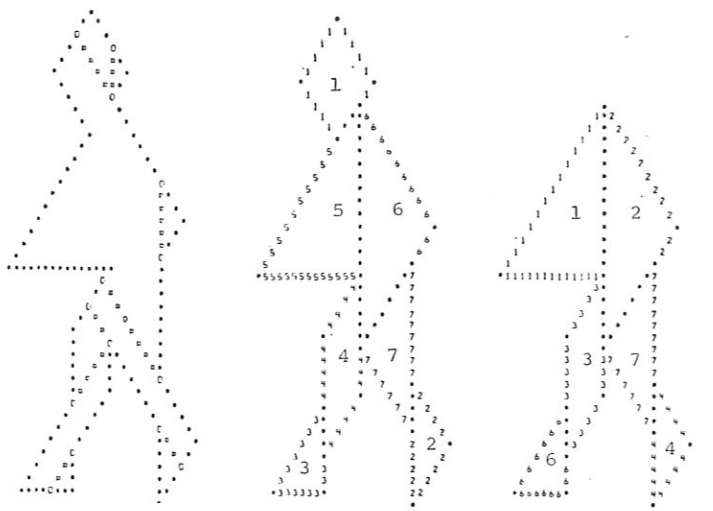
4.7 Puzzle pieces 2-3 edge-match rule

We now return to further rules for extraction of individual puzzle pieces. The rule under this heading is *the same as that just developed for composite pieces.* This rule never fails to extract the four-sided puzzle pieces correctly, but it can introduce errors when it tries to isolate the smaller triangular pieces. The rule is designed to isolate the small triangular pieces only if no other piece can be isolated using this rule. Figure 12 shows a complete tangram solution obtained using the $2\frac{1}{2}$–$3\frac{1}{2}$ edge rule and the 2–3 edge rule. (The example introduces the need for a good organization of the rules.) The larger triangular piece in figure 12(i) is identified by the 2–3 edge rule.

Figure 13 shows the solution to a tangram puzzle using the $2\frac{1}{2}$–$3\frac{1}{2}$ edge-match rule and the 2–3 edge-match rule. It illustrates the need for the restriction concerning small triangles in the latter rule. Observe that in figure 13(c) the small triangles, remaining after the $2\frac{1}{2}$–$3\frac{1}{2}$ has been applied, could



(a) (b) (c)



(d) (e) (f)

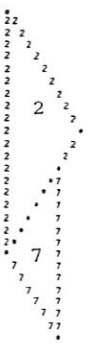
Figure 12

be fitted into areas ABC or DE
Now that the small triangles a

4.8 Puzzle pieces: $1\frac{2}{2}$-$2\frac{2}{2}$ edge-

This rule is one of the most l
puzzle piece to be recognized
the remaining sides can be de
lines. The application of the ru
puzzle piece in figure 14 is id

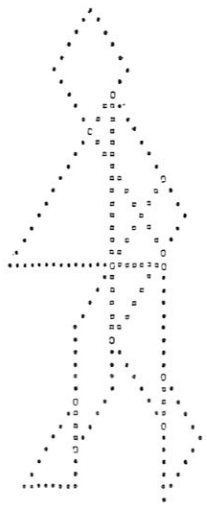
(g)



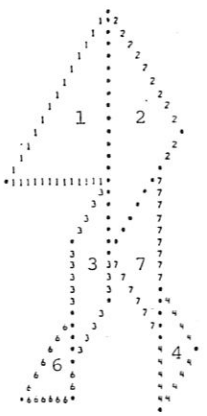
(j)

Figure 12 contd.

uzzle using the $2\frac{1}{2}$–$3\frac{1}{2}$ edge-
ates the need for the restric-
ule. Observe that in figure
$-3\frac{1}{2}$ has been applied, could



(c)

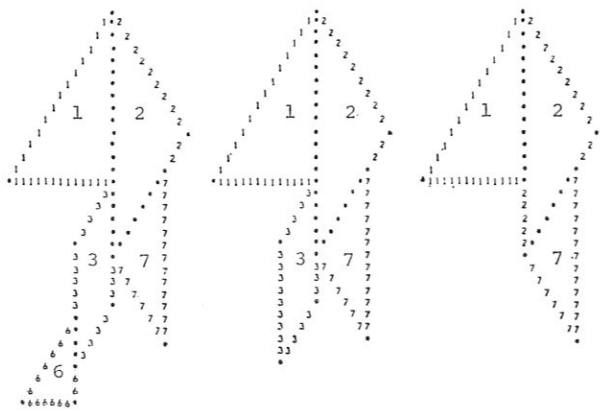


(f)

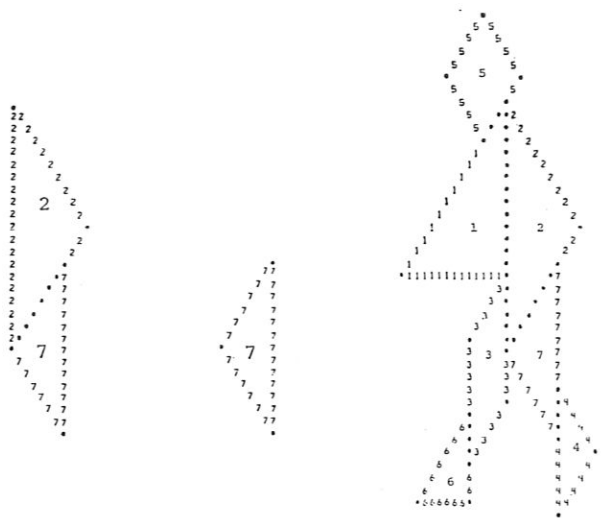
be fitted into areas ABC or DEF. The former choice would lead to an error. Now that the small triangles are matched last this will not occur.

4.8 Puzzle pieces: $1\frac{2}{2}$-$2\frac{2}{2}$ edge-match rule

This rule is one of the most lax. All that will be required for a three-sided puzzle piece to be recognized is that only *one of its sides consist of an edge; the remaining sides can be defined by a combination of edges and extension lines.* The application of the rule to four-sided puzzle pieces is clear. The first puzzle piece in figure 14 is identified using this rule (see figure 14(c)).

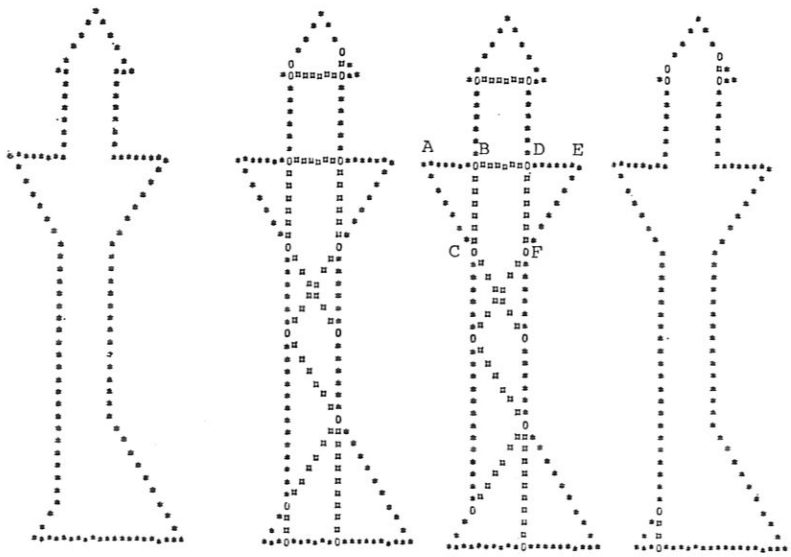


(g) (h) (i)

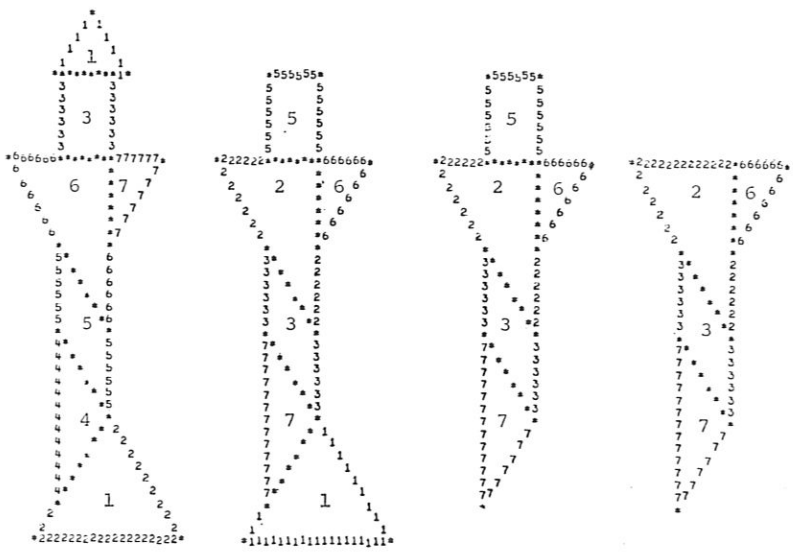


(j) (k) (l)

Figure 12 contd.

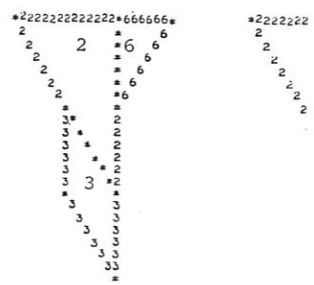


(a) (b) (c) (d)



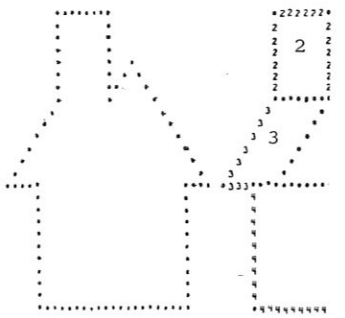
(e) (f) (g) (h)

Figure 13

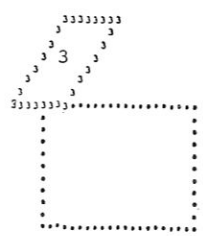


(i)

Figure 13 contd.



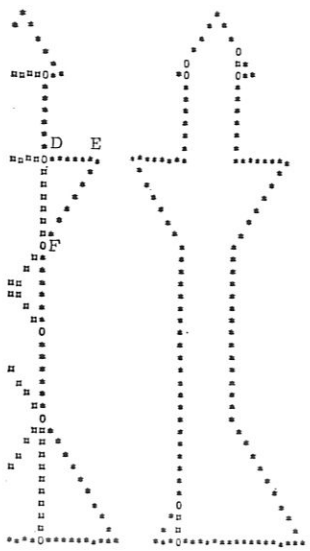
(a) (



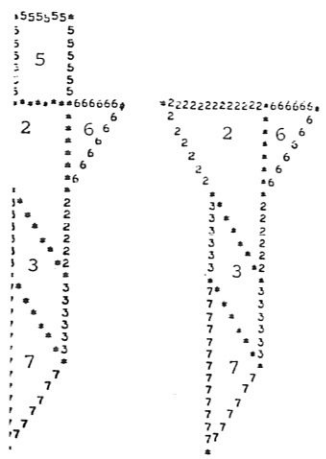
(e)

Figure 14

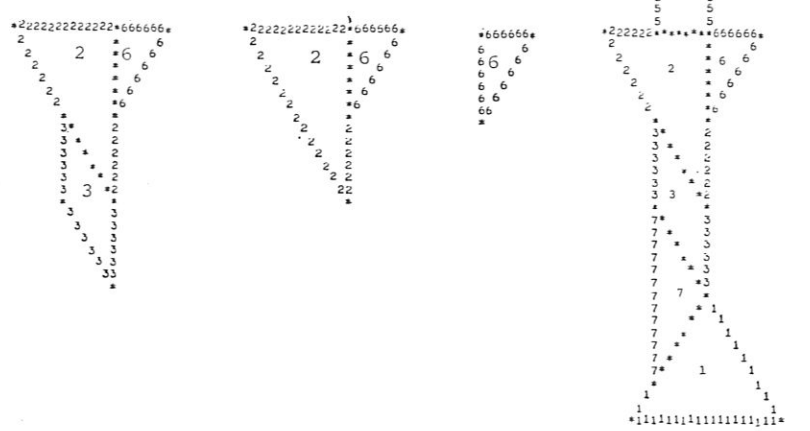
Q



(c) (d)

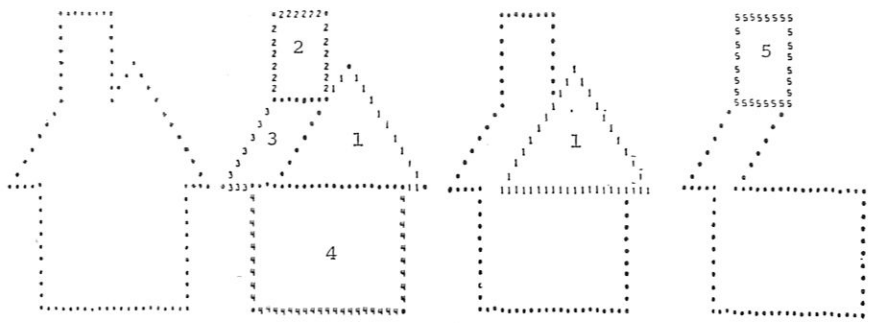


(g) (h)

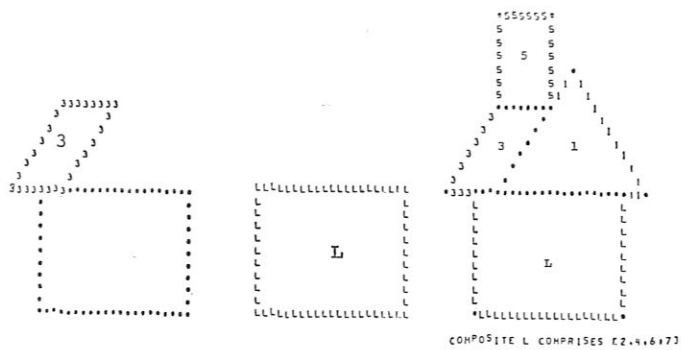


(i) (j) (k) (l)

Figure 13 contd.



(a) (b) (c) (d)



(e) (f) (g)

Figure 14

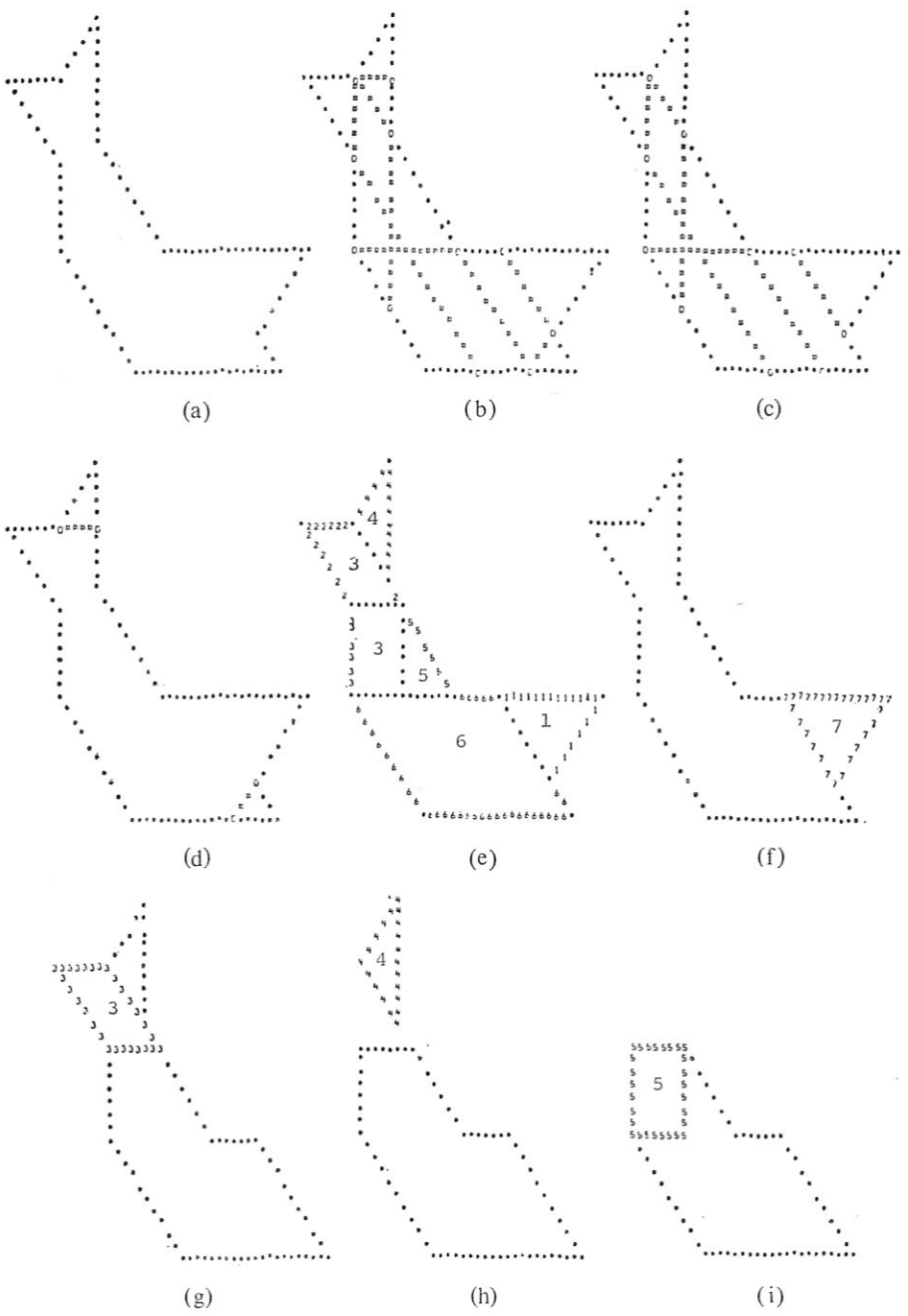


Figure 15

4.9 Puzzle pieces: 3 edge onl

This and the next rule appl
pieces only. The rule require
sides, each being defined by
tangram of figure 14 emplo
figure 14 has changed somew
pieces are extracted. The e
order in which correspondin

4.10 Puzzle pieces: 2 edge a

This rule, too, requires the *f*
their sides; here, however, *o*
Figure 15 shows a solution t

4.11 Puzzle pieces: 2 edge o

This rule applies to the extra
only *two sides to be defined b*
a 'last resort' rule since it is
solution. Figure 16 shows a
solution process. Observe t
16(c) indicates that the bott
indeed extracted then, but
should not be confused wi
triangle (also of piece numb

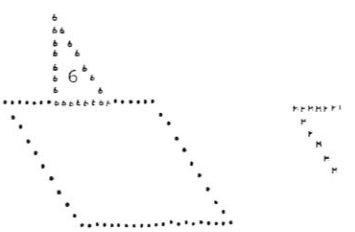
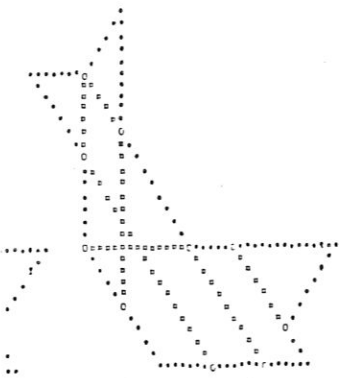


Figure 15 contd.



(c)

(f)

(i)

4.9 Puzzle pieces: 3 edge only match rule

This and the next rule apply to the extraction of the two *four-sided* puzzle pieces only. The rule requires these puzzle pieces to be *represented by three sides, each being defined by an edge*. The latter part of the solution to the tangram of figure 14 employs this rule (see figure 14(e)). The format of figure 14 has changed somewhat now to conform to the actual way in which pieces are extracted. The edge numbers in figure 14(b) still refer to the order in which corresponding pieces are removed.

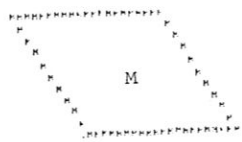
4.10 Puzzle pieces: 2 edge and 1 extension line match rule

This rule, too, requires the *four-sided* puzzle pieces to be defined by *three of their sides*; here, however, *one* side can consist entirely of an extension line. Figure 15 shows a solution to a puzzle employing this rule.

4.11 Puzzle pieces: 2 edge only match rule

This rule applies to the extraction of the *two large* triangles only. It requires only *two sides to be defined by edges. The third side need not exist*. This rule is a 'last resort' rule since it is evoked, usually, after no other rule provided a solution. Figure 16 shows an example where this rule is used to initiate a solution process. Observe that no extension lines are deleted here. Figure 16(c) indicates that the bottom portion of the left limb is solved second. It is indeed extracted then, but its identity is only determined at the end. This should not be confused with figure 16(e) which indicates that the large triangle (also of piece number 2) is identified second.

(j)



(k)

(l)

Figure 15 contd.

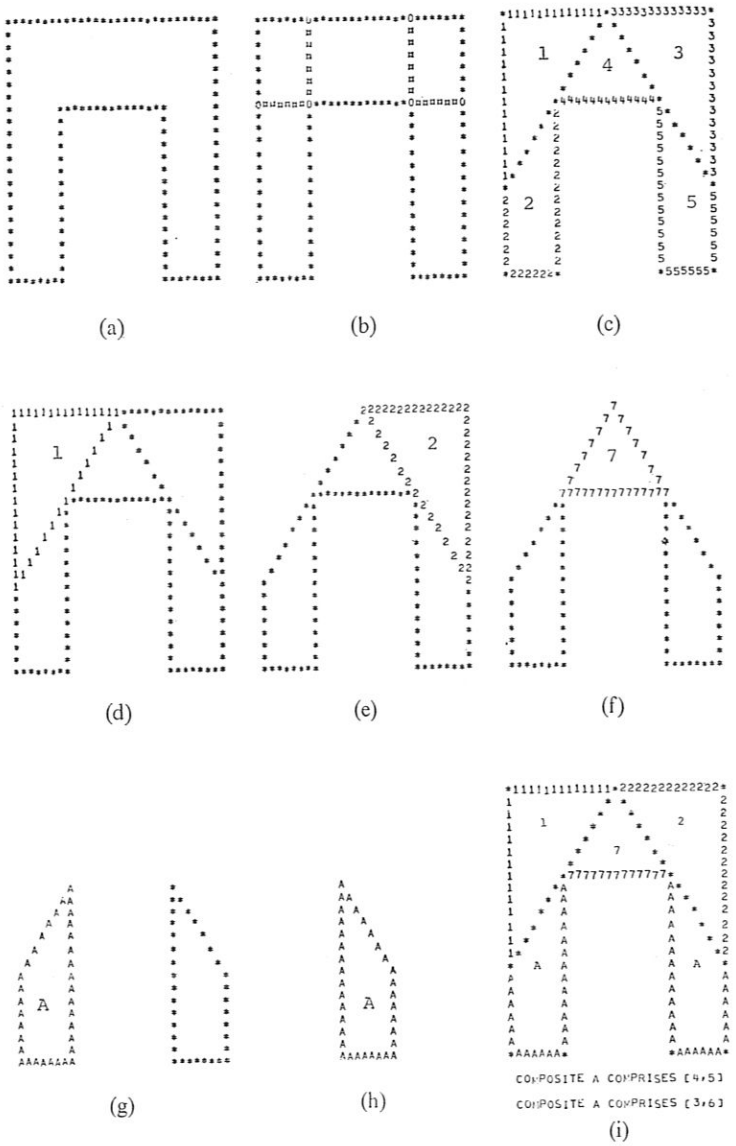


Figure 16

5. ORGANIZATION OF MATCH RULES

The tangram solutions presented in the above sections were derived by applying the puzzle piece extraction rules according to the scheme shown in figure 17. This figure should be examined together with figure 2. Some preliminary ranking is necessary in order to determine the efficacy of the

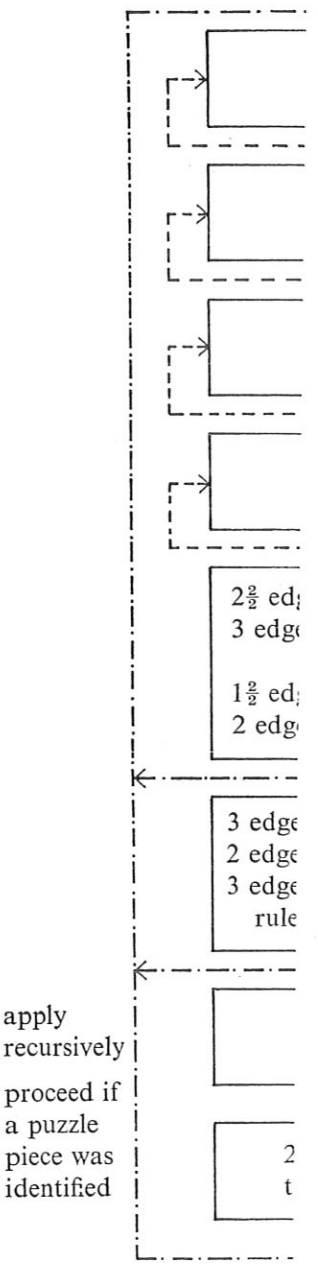
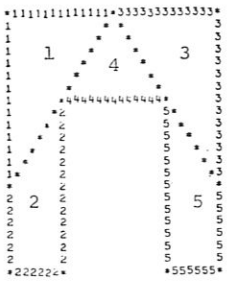
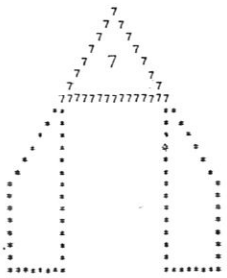


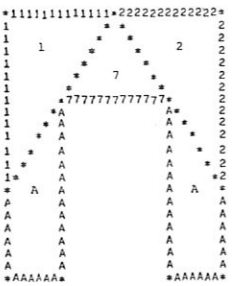
Figure 17



(c)



(f)



(i)

ES

ove sections were derived by
ording to the scheme shown in
ogether with figure 2. Some
determine the efficacy of the

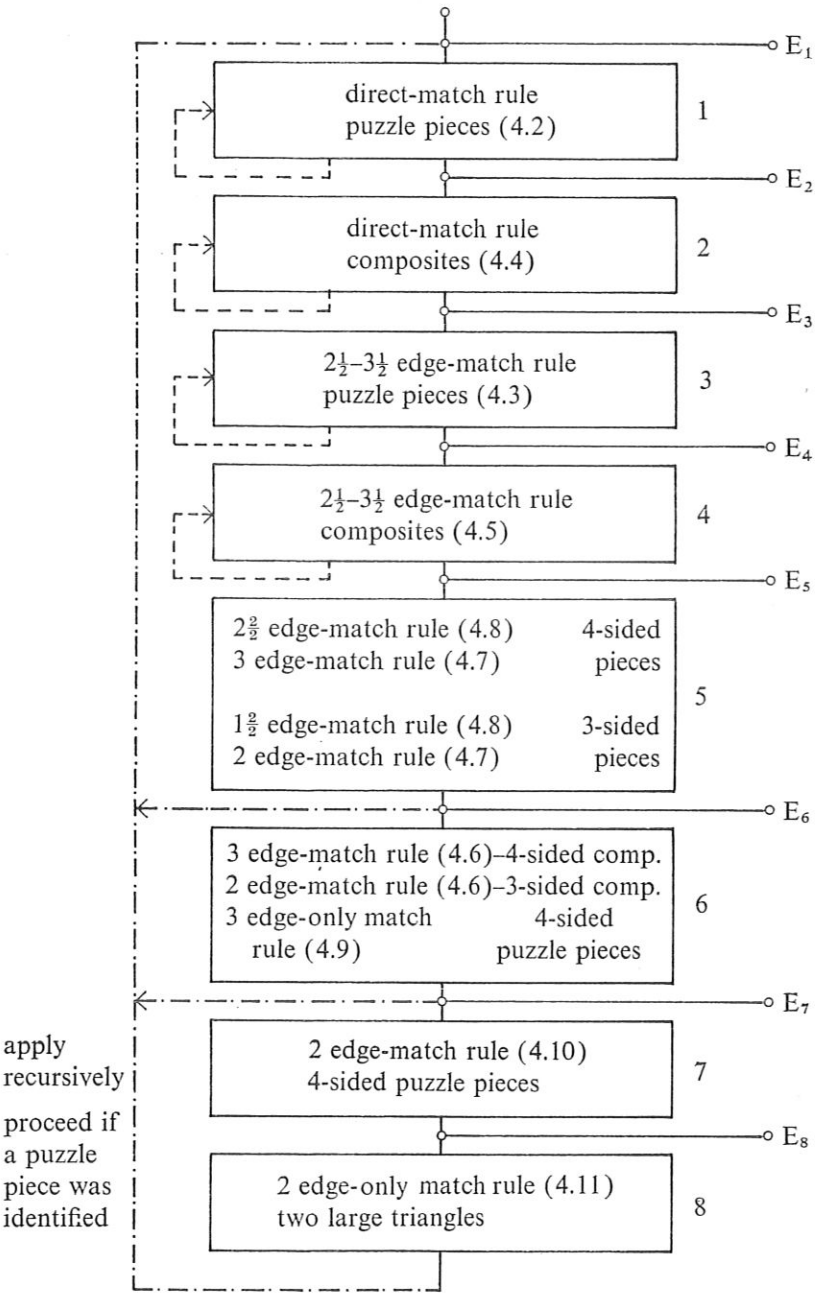
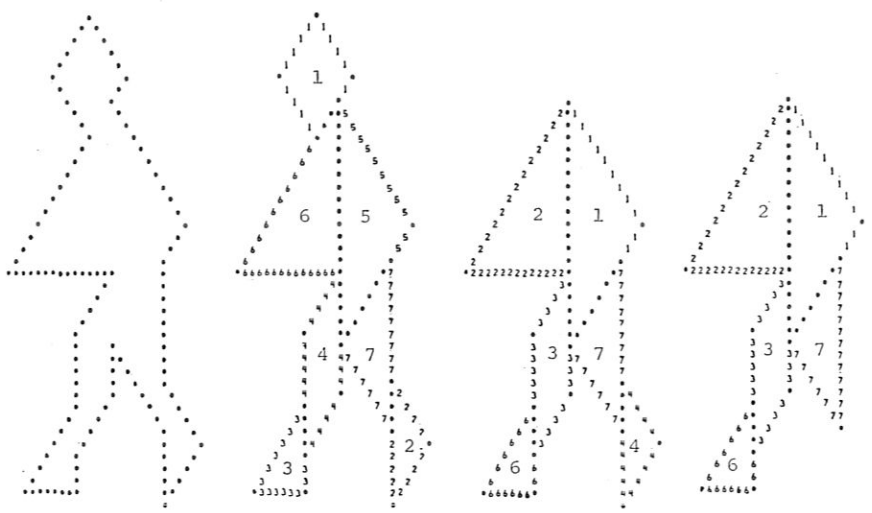
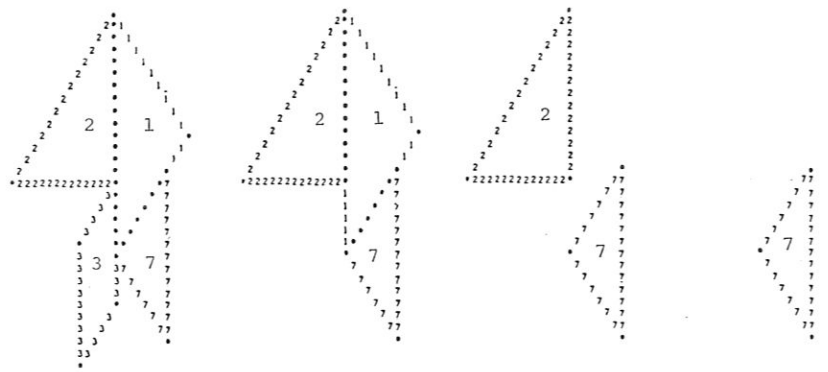


Figure 17

proposed extraction rules. By the scheme of figure 17, the first four rules are each applied recursively, that is, the first rule is applied until no further pieces (if any) are extractable, and is then followed by the second rule also applied recursively, and so on up to the fifth rule. The fifth rule is applied once and should a puzzle piece have been recognized an immediate return to the first rule is made. Failing this, the sixth extraction rule is applied. Entry points are declared at the beginning of the program. The reason underlying this preliminary organization is that the first four extraction rules are considered 'cast-iron' rules and should therefore be applied as often as possible.



(a) (b) (c) (d)



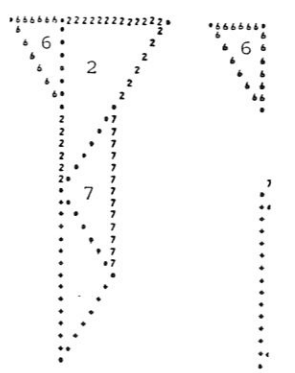
(e) (f) (g) (h)

Figure 18

The rules of lower rank, ho
therefore be applied only wh
Once the solution process is
be resorted to again. The qu
the rules.

Most of the reorganizatio
the lower portion of the diag
interesting to see what hapj
rules is changed. Figure 18

(a)



(e)

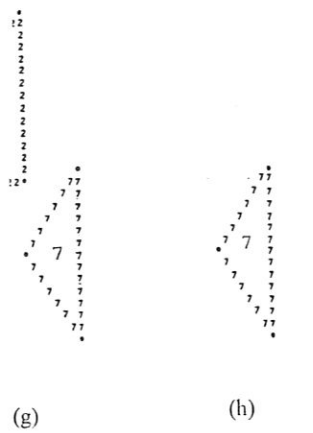
Figure 19

', the first four rules are
pplied until no further
by the second rule also
The fifth rule is applied
l an immediate return to
ɔn rule is applied. Entry
.. The reason underlying
extraction rules are con-
ɔlied as often as possible.

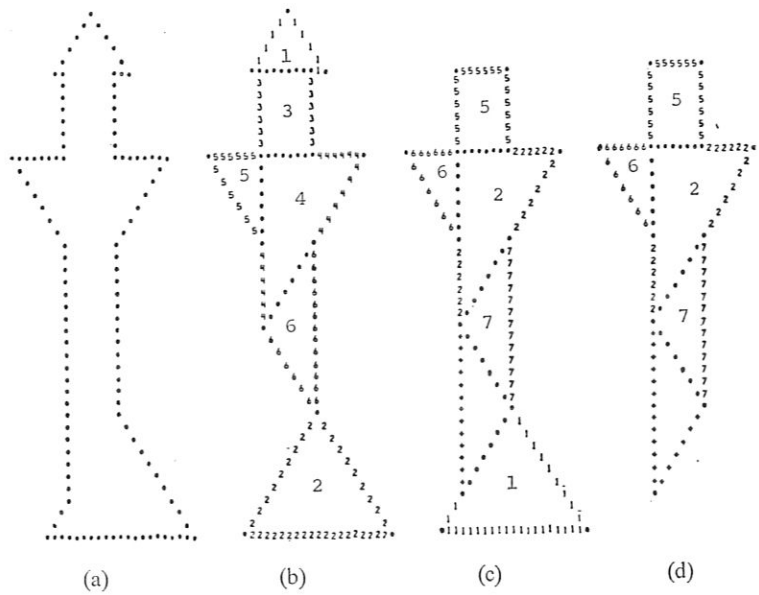
The rules of lower rank, however, are increasingly less reliable and should therefore be applied only when the solution process can proceed no further. Once the solution process is continued more reliable extraction rules should be resorted to again. The question is what is the most efficient way to rank the rules.

Most of the reorganization of the rules in figure 17 will take place within the lower portion of the diagram, since the first few rules seem reliable. It is interesting to see what happens to a tangram solution when the order of rules is changed. Figure 18 shows the solution of the tangram of figure 12.

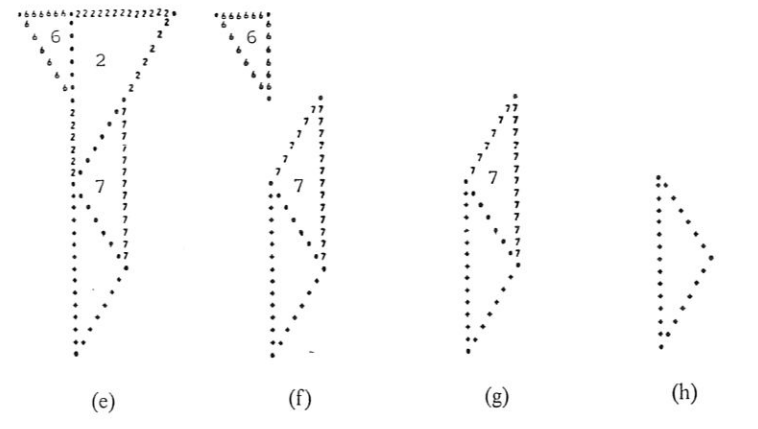
:) (d)



(g) (h)



(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 19

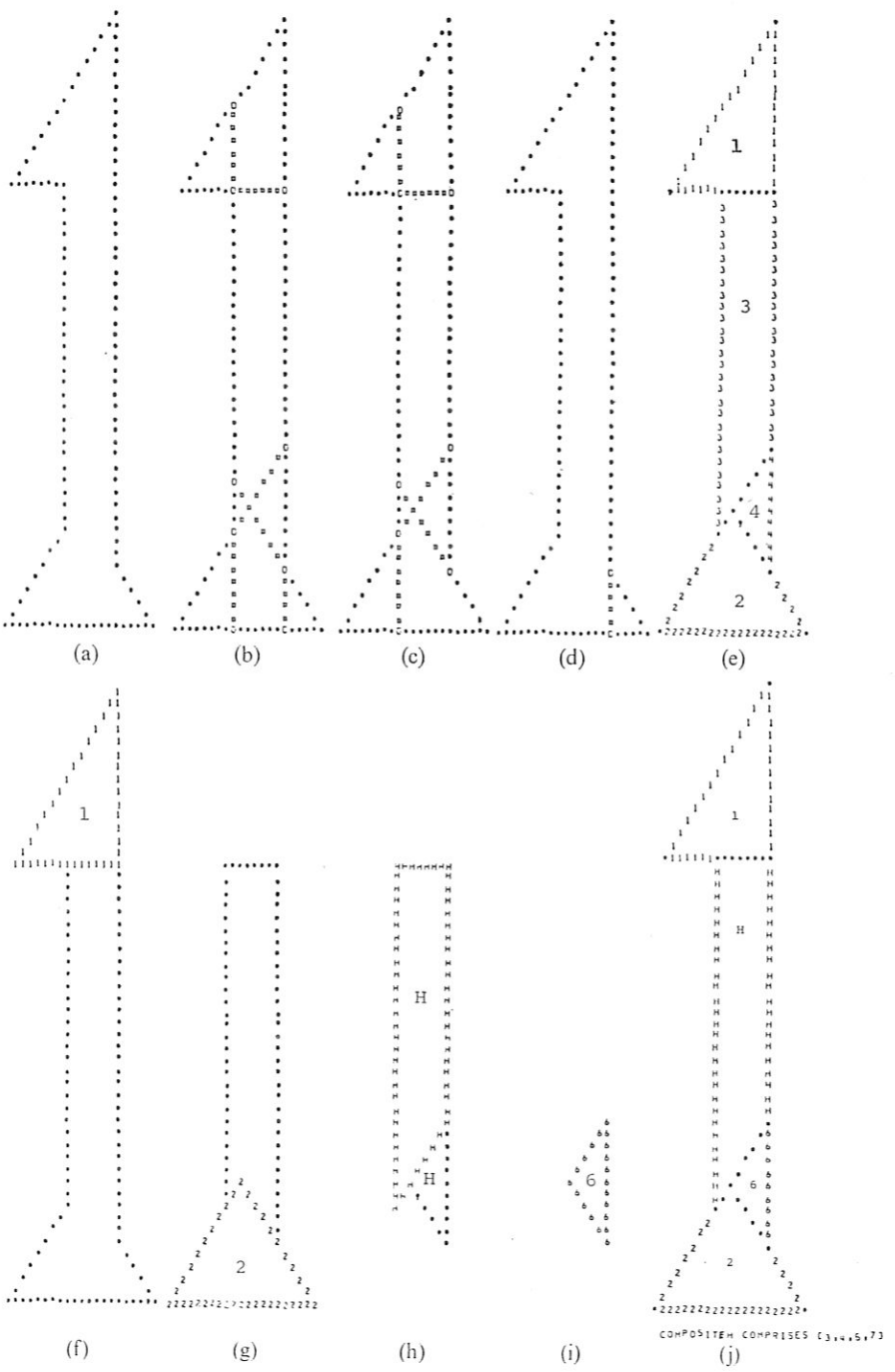


Figure 20

It will be observed on compa
puzzle pieces are extracted is
ceded the $1\frac{2}{2}$-edge rule in deri
A correct solution is obtaine
order of rules to the tangran
shown in figure 19. Here t
triangles. Observe that the o
are used to solve the tangram

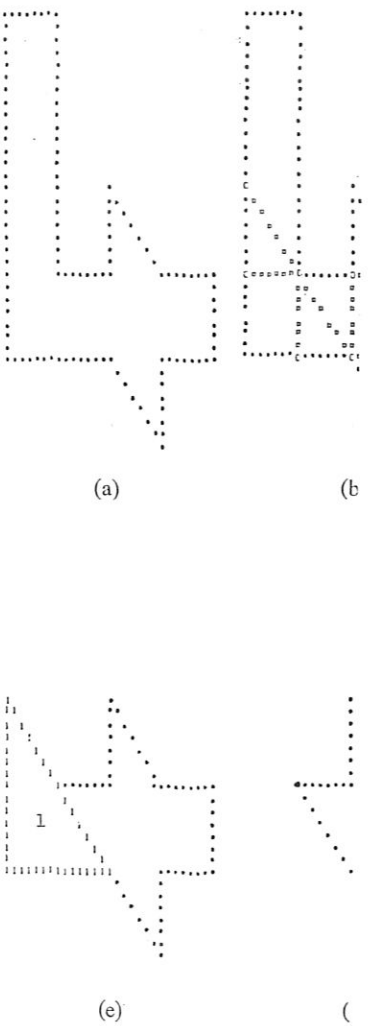
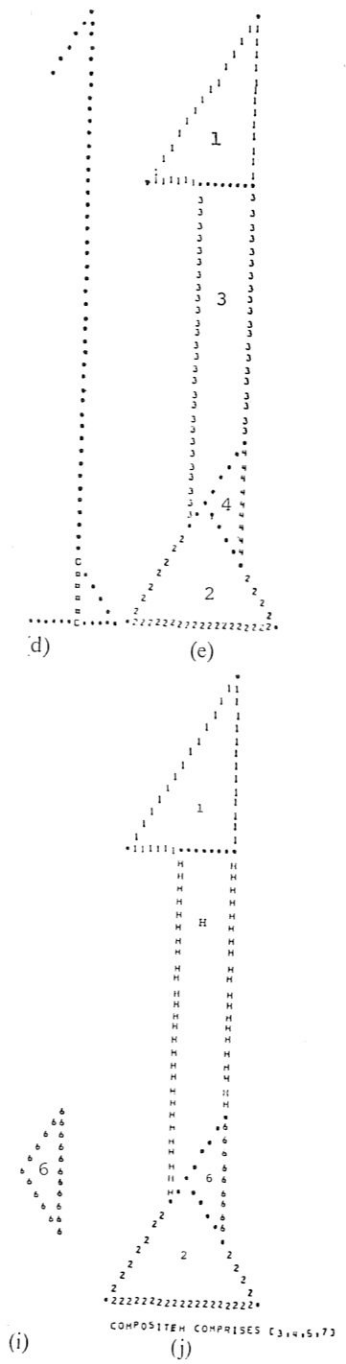


Figure 21

It will be observed on comparing the two solutions that the order in which puzzle pieces are extracted is different. This is because the 2-edge rule preceded the $1\frac{2}{2}$-edge rule in deriving the latter solution (see box 5 in figure 17). A correct solution is obtained either way in this case, but applying the new order of rules to the tangram of figure 13 produces an erroneous result as shown in figure 19. Here the program tries to elicit two medium-sized triangles. Observe that the old format is used here. The rules of figure 17 are used to solve the tangram of figure 9(b), shown in figure 20.



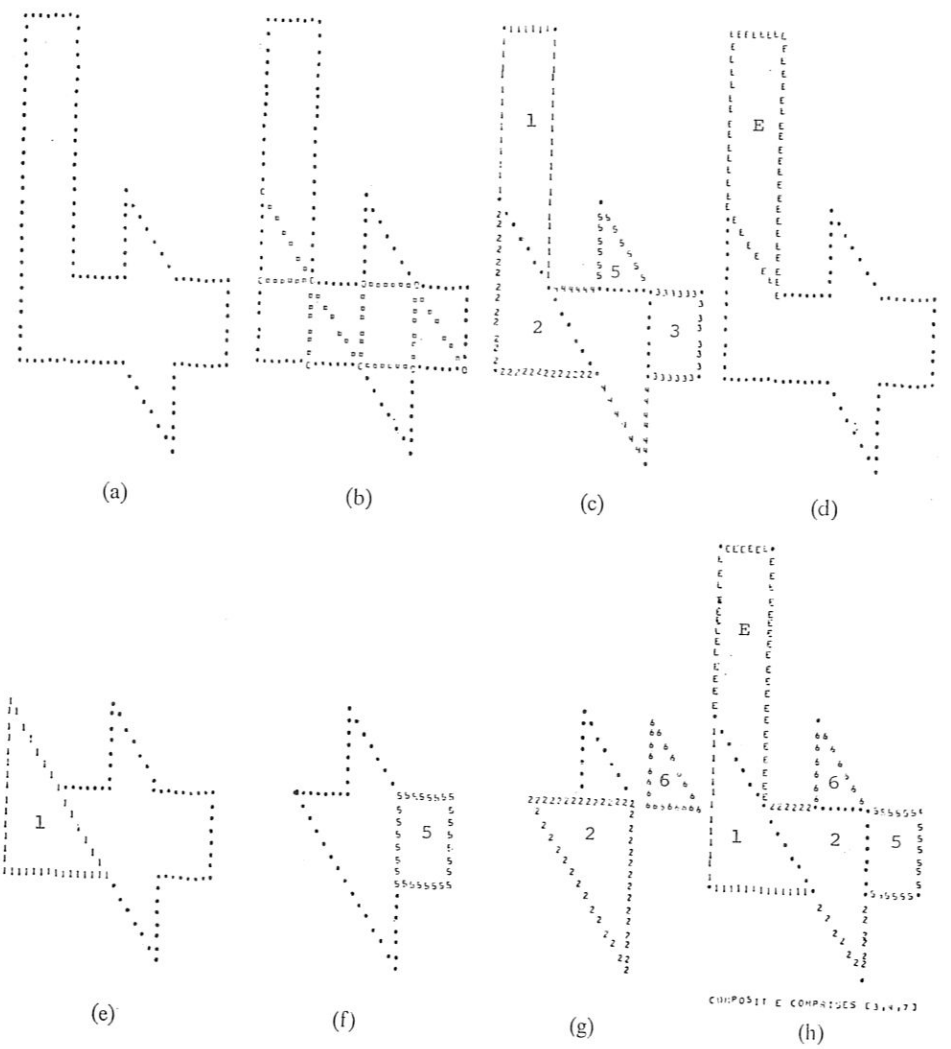


Figure 21

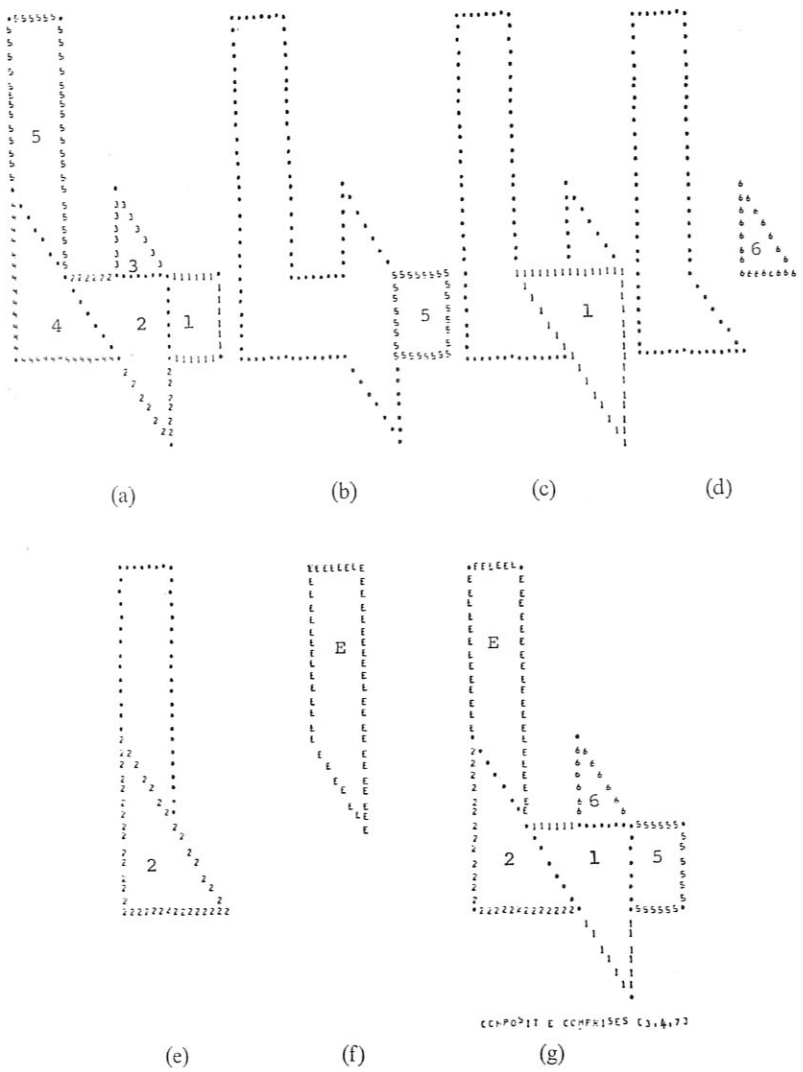


Figure 22

Figure 21 and figure 22 show the difference in the way a solution is obtained when the priorities of the rules for the extraction of puzzle pieces and the rules for the extraction of composites are interchanged. In general, if the rules for the extraction of puzzle pieces are given a higher priority, then the solution will occur earlier. When a composite piece is extracted earlier, the number of attempts at a solution will increase. Figure 23 gives the flow chart of the way the rules are finally organized. The chart consists of two parts:

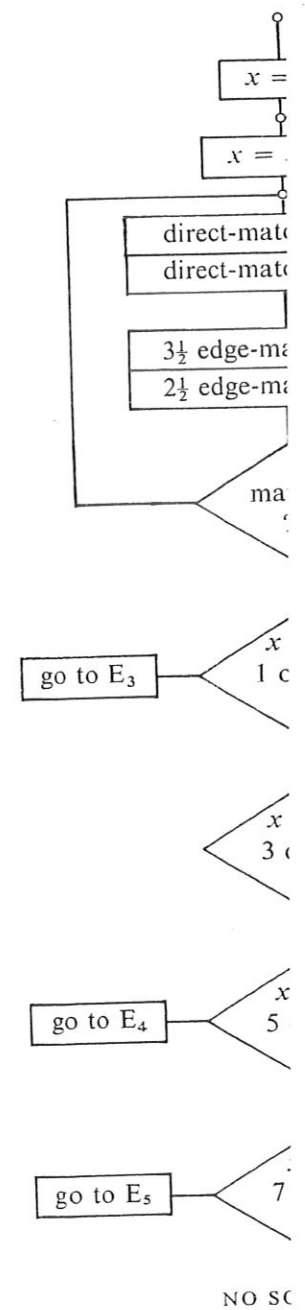
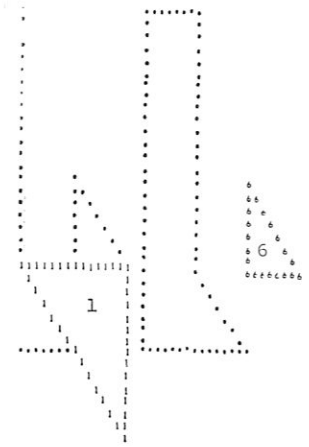
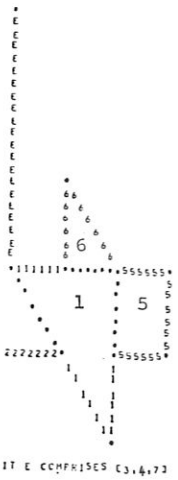


Figure 23(a)



(c) (d)



(g)

the way a solution is obtained
ion of puzzle pieces and the
erchanged. In general, if the
en a higher priority, then the
piece is extracted earlier, the
Figure 23 gives the flow chart
chart consists of two parts:

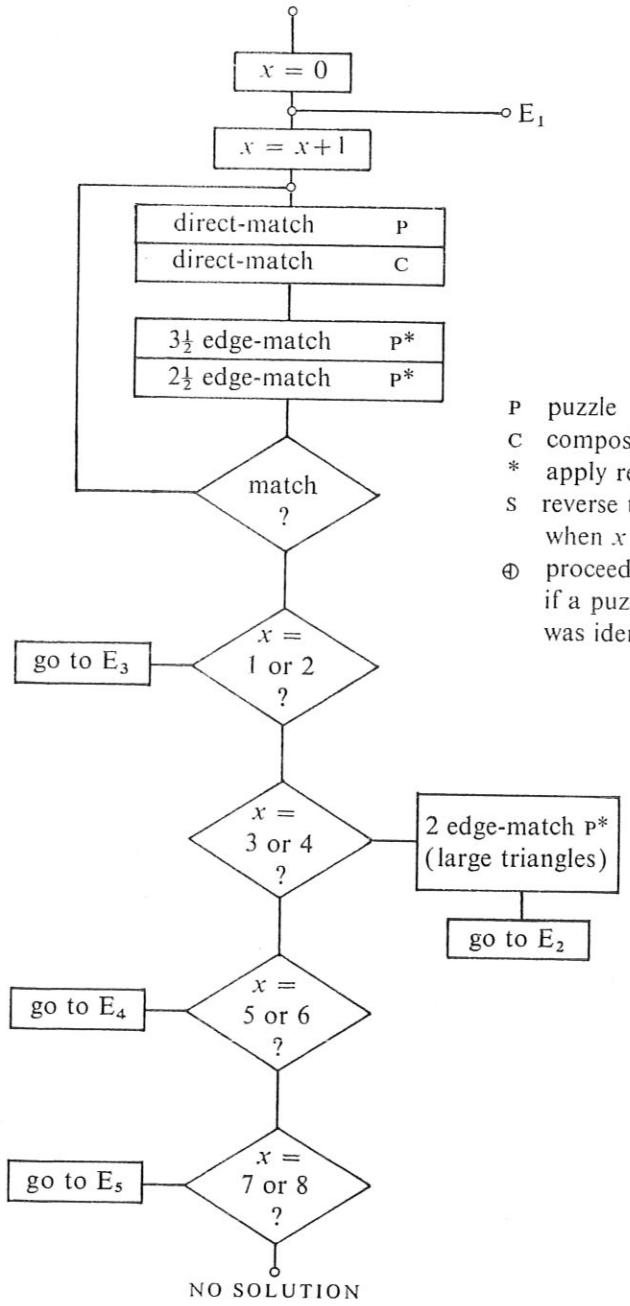


Figure 23(a)

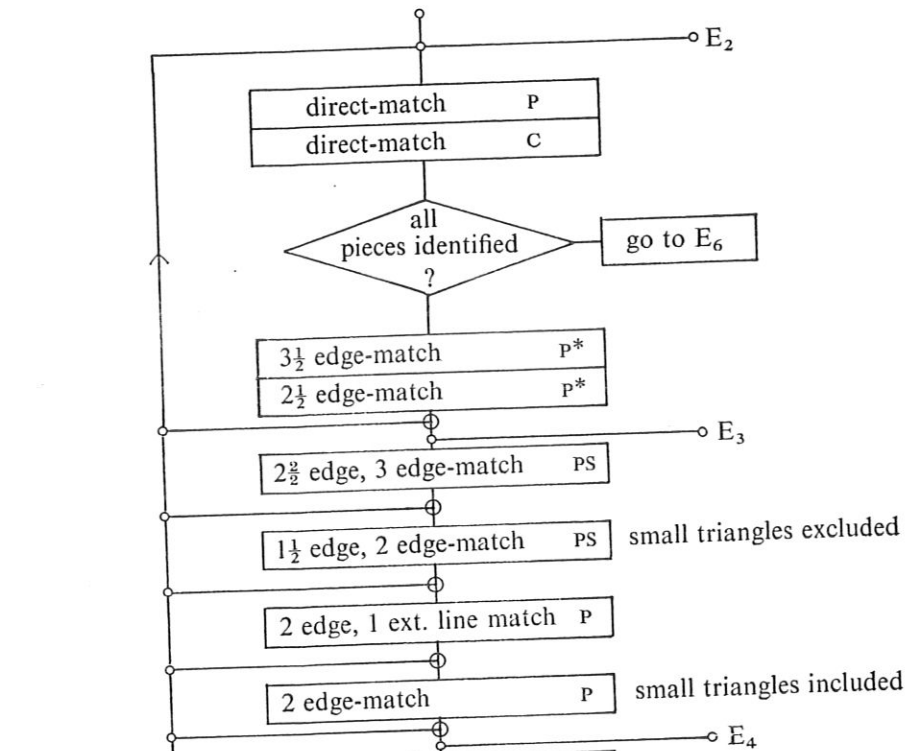


Figure 23(b)

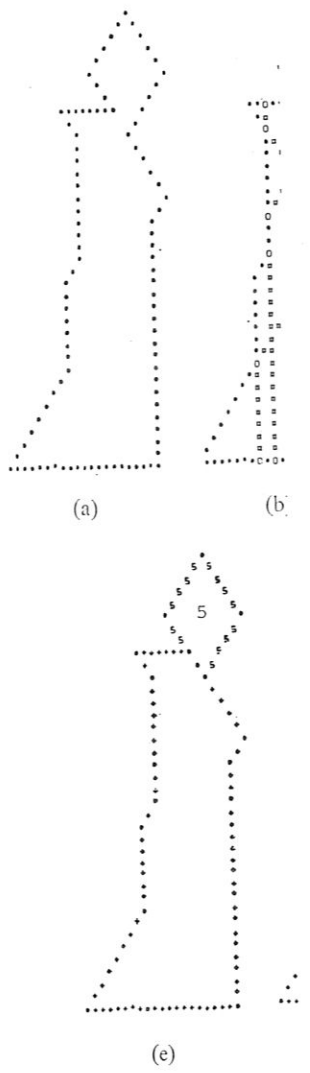
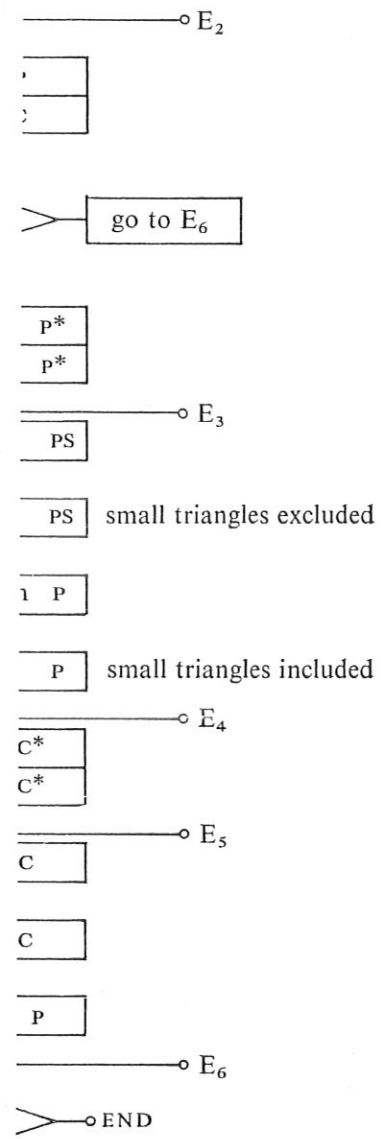


Figure 24

the main program, controll
applied, and the remaining
Considerations of space
figure 24(a) shows a tangr
The partial solutions obtair
also shown.



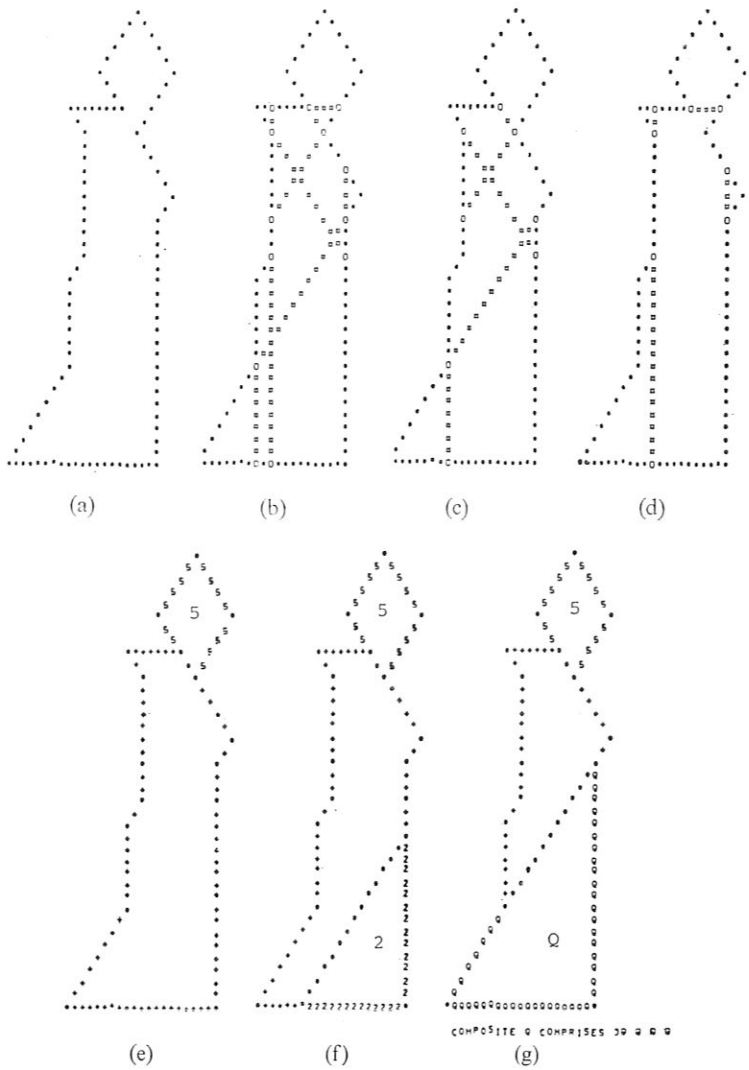


Figure 24

the main program, controlling the order in which the extraction rules are applied, and the remaining part consisting of the rules themselves.

Considerations of space prevent the showing of further results. However, figure 24(a) shows a tangram puzzle for which a solution was not found. The partial solutions obtained at various stages of the solution process are also shown.

CONCLUSION

It has been shown how a tangram puzzle can be solved using some very simple extraction rules. The rules are based upon the location of edges which define individual puzzle pieces and composites of puzzle pieces. The edges are either those defined by the actual puzzle or edges constructed subsequently. The less edges required by a given extraction rule the less reliable this rule is. This leads to the method by which the rules were ranked.

In principle, the approach used here is applicable to any form of puzzle, in which, like the tangram puzzle, the shape of individual puzzle pieces is simple, and their number is restricted. The material presented here could well be used with an interactive scheme in which the user can specify the particular rule he wishes to apply to a puzzle during a solution process.

Acknowledgements

The support of the National Science Foundation, under Grant GJ–754, is gratefully acknowledged.

REFERENCES

Freeman, H. & Garder, L. (1964) Apictorial jigsaw puzzles: the computer solution of a problem in pattern recognition. *IEEE Trans. EC–13*, 118–27.

Guzman, A. (1968) Decomposition of a visual scene into three-dimensional bodies. *Proc. AFIPS 1968 Fall Joint Comp. Conf.*, **33**, 291–304. Washington DC: Thompson Book Co.

Read, R.C. (1965) *Tangrams*. New York: Dover Publications Inc.

13

A Man-Machine Solutions to Urb

P. D. Krolak and J. H

Vanderbilt University
Nashville, Tennessee

Abstract

The problems of the urban ar inter-related. Often they invol classical techniques of mathe are frequently unsatisfactory. solely by human subjective ju the scope of the problems is makers. This paper outline problem solving which makes An information hierarchy is d areas of heuristic problem so world problems, garbage co achieve a racial balance, ar future directions for research

INTRODUCTION

During the last decade a nu toward the area of problem Newell and Simon (1972), th to be formulated. Michie (19 1970, Kilgour 1971) have s efficient it is in solving proble (Feigenbaum and Feldman 1 require intelligence, that is, 1 (2) attempt to simulate hu discusses another aspect of pr the ability of the man-machi