Computer Science, Duke Univ., Durham, NC, Rep. CS-1976-4, June 1975.

[3] ——, "Approaches to automatic programming," in *Advances in Computers*, vol. 15, M. Rubinoff and M. Yovits, Eds. New York: Academic, 1976, pp. 1-63.

[4] A. W. Biermann, R. I. Baum, and F. E. Petry, "Speeding up the synthesis of programs from traces," *IEEE Trans. Comput.*, vol. C-24, no. 2, pp. 122-136, 1975.

[5] A. W. Biermann and R. Krishnaswamy, "Constructing programs from example computations," *IEEE Trans. Software Eng.*, vol. SE-2, Sept. 1976.

[6] L. Blum and M. Blum, "Toward a mathematical theory of inductive inference," *Inform. Contr.*, vol. 28, pp. 125-155, 1975.

[7] J. A. Feldman, "Some decidability results on grammatical inference and complexity," *Inform. Contr.*, vol. 20, no. 3, pp. 244-262, 1972.

[8] M. Gold, "Language identification in the limit," *Inform. Contr.*, vol. 10, no. 5, pp. 447-474, 1967.

[9] C. C. Green *et al.*, "Progress report on program understanding systems," Stanford Artificial Intelligence Laboratory, Stanford, CA, Memo AIM-240, 1974.

[10] S. Hardy, "Synthesis of LISP functions from examples," in *Advance Papers 4th Int. Joint Conf. Artificial Intelligence*, Tbilisi, Georgia, USSR, Sept. 1975, pp. 240-245.

[11] J. McCarthy, P. W. Abrahms, D. J. Edwards, T. P. Hart, and M. I. Levin, *LISP 1.5 Programmer's Manual*. Cambridge, MA: M.I.T., 1962.

[12] D. Shaw, W. Swartout, and C. Green, "Inferring LISP programs from examples," in *Advance Papers 4th Int. Joint Conf. Artificial Intelligence*, Tbilisi, Georgia, USSR, Sept. 1975, pp. 260-267.

[13] L. Siklóssy and D. A. Sykes, "Automatic program synthesis from example problems," in *Advance Papers 4th Int. Joint Conf. Artificial Intelligence*, Tbilisi, Georgia, USSR, Sept. 1975, pp. 268-273.

[14] P. D. Summers, "Program construction from examples," Ph.D. dissertation, Yale Univ., New Haven, CT, 1975.

[15] ——, "A methodology for LISP program construction from examples," *J. Ass. Comput. Mach.*, vol. 24, pp. 161-175, 1977.

Arrangements, Homomorphisms, and Discrete Relaxation

ROBERT M. HARALICK, SENIOR MEMBER, IEEE, AND JESSE S. KARTUS

Abstract—**We show how homomorphisms between arrangements, which are labeled N-ary relations, are the natural solutions to some problems requiring the integration of low-level and high-level information. Examples are given for problems in point matching, graph isomorphism, image matching, scene labeling, and spectral temporal classification of remotely sensed agricultural data. We develop characterization and representation theorems for N-ary relation homomorphisms, and we develop an algorithm consisting of a discrete relaxation method combined with a depth-first search to find such homomorphisms.**

I. Introduction

The usual procedure we follow when we wish to analyze a complex structure is to divide up the world into simple and separate atomic units, to observe or measure some basic properties of these units, and then to use the measured properties to name or describe the pattern among the units. Although this protocol is effective and powerful for simple structures, it has inherent problems for complex structures; namely, the measured units may not be separate and independent, the low level measurements may be noisy since they are made locally without the benefit of any system integration, the units themselves may have been chosen more for the convenience of the measurement-taking process than for their importance in analyzing the structure in which we are really interested, and finally, we might not have an effective mathematics which facilitates the graceful incorporation of low-level information into high-level information. In this paper we describe how the concept of arrangements is applicable to some of these problems. We show how arrangements and arrangement homomorphisms provide a natural perspective and method by which information can be compared and by which known *a priori* information can be used gracefully in integrating micro and macro knowledge of a structure. We illustrate that a variety of particular problems constitute the same mathematical/combinatorial problem, and we give an algorithm to solve the general mathematical problem.

Section II defines the arrangement concept. Section III discusses the similarity between order-N arrangements and defines arrangement homomorphisms. Section IV gives some examples of problems which require the finding of homomorphisms from one arrangement to another. Section V gives an algorithm for finding arrangement homomorphisms using the winnowing relaxation process combined with a depth-first search. The Appendix gives the formal statements and proofs for the assertions made in Section V.

II. The Arrangement

Let $S = \{s_1, \cdots, s_K\}$ be a set of K possible descriptions of measurements that could be given to a unit. Each unit is given none, one, or more measurement descriptions. For

Manuscript received November 1, 1976; revised May 9, 1977 and March 24, 1978.

The authors are with the Image Processing Laboratory, Space Technology Center, University of Kansas, Lawrence, KS 66045.

example, in an image interpretation problem, the units could be resolution cells, and the set S could be the possible directions of an edge passing through the resolution cell. The units could also be image segments, and the set S could consist of quantitative or qualitative measures of the shapes of the areal segments. In an urban geographical problem, the units might be neighborhoods, and the set S could consist of different land use types. In an abstract mathematical situation, the units could be points in an N-dimensional Euclidean space, and the set S could consist of the ordered N-tuple of point coordinates.

In each one of these examples, as in general, the measuring instrument or sensor provides a description, value, or label to each unit based only on the unit itself and not on the unit's relationship to other units. Hence, the simplicity of the independence of the atomic units is maintained for this first low-level measurement-taking process.

In order to determine the pattern among the units, descriptions of related units must be considered together. We examine relevant combinations of related units taken N at a time and associate a set of possible interpretations with each combination. This constitutes a second-level naming process which depends on a given specified relation among the units. The language used in this process can be different from the values or phenomenological language used in the initial measurement-taking stage. The number N indicates the order of complexity we are willing to examine. For example, suppose we wish to examine units which are points in some inner product space up to a degree of complexity equal to three. We could use the point coordinates values for the first-level measurement description and then use the three angles of a triangle determined by any three points as the second-level names.

There are some differences between the first-level measuring process and the second-level describing process. In the first level, each unit is considered by itself and measured independently of other units around it. In the second-level describing process, units are considered in specially related groups of size N. Not necessarily all groups or combinations of size N need be considered, only those considered relevant by the investigator. In the first-level describing process, each unit is given none, one, or more descriptions. In the second-level describing process, any relevant group of units can be given one or more interpretations, and the language of the interpretations can be entirely different from the "sense-data" language of the initial description.

The second-level interpreting process specifies an $(N + 1)$-ary relation F. If S is the first-level set of descriptions and D is the second-level set of interpretations, then we may define the arrangement F as a subset of the Cartesian product of S, N times, with the Cartesian product of D:

$$F \subseteq \underbrace{S \times S \times \cdots \times S}_{N \text{ times}} \times D.$$

An Nth-order arrangement is really a generalization of some familiar mathematical structures. For example, a binary relation is a second-order arrangement with all pairs named the same. A labeled graph is a second-order arrangement with the ordered pairs having a variety of labels. An automaton is also a second-order arrangement. An automata $\mathscr{A}$ is usually defined as a triple (S, Σ, δ) where $\delta \subseteq S \times \Sigma \times S$. When δ is a function from $S \times \Sigma$ into S, the automaton is completely specified and deterministic. When δ is a relation, the automaton may be incompletely specified or nondeterministic. By interchanging the second two components of the relation δ we have $\delta \subseteq S \times S \times \Sigma$, and we see that the automaton $\mathscr{A}$ is a second-order arrangement. The difference between the general automaton (incompletely specified and nondeterministic) and the second-order arrangement is that a sequential interpretation is put on the labeled order pairs of the automaton; each labeled ordered pair is the transition of a state to another state under a particular input. In the second-order arrangement, each ordered pair just has a name, there is no from–to interpretation. In the third-order arrangement, there can be no from–to interpretation, and the name is just a name for the triple.

Barrow, Ambler, and Burstall [1] suggest using labeled N-ary relations for organizing structural information in image analysis. The parametrized structural representation of Hayes-Roth [4] is easily translated to a set of arrangements. The relational data base [2] consisting of relational tables is closely related to the arrangement structure. The MSYS system for scene analysis at Stanford Research Institute uses a representation scheme related to the arrangement structure [6]. Minsky [10] seems to hint at an arrangement structure in discussing "frames." Hanson and Riseman's [11] region segment endpoint relations and their frames, objects, and surface relations can be related to arrangements.

III. The Similarity Between Two Order-N Arrangements

Consider an order-N arrangement as a pattern. To classify order-N arrangements from the pattern recognition point of view entails finding a decision rule which will assign one of several category labels to an order-N arrangement based on the similarity the order-N arrangement has with the category prototype arrangement. In parametric pattern recognition, it is common to begin in a metric space and base the similarity between two patterns on the distance between them. Then a probability that a vector is generated by a category can be defined as a function of a generalized distance measure between the vector and a prototype vector such as the mean of the category. For example, the multivariate normal distribution is one distribution in the class of ellipsoidally symmetric distributions all of whose point densities are monotonically decreasing functions of the Mahalanobis distance between the point and the distribution mean.

It is not as easy to define a meaningful metric space on the set of order-N arrangements as it is to do so in a vector space. In this paper, we suggest approaching the idea of similarity between two order-N arrangements algebraically, using homomorphisms. So we need to find a way of determining all the homomorphisms of one arrangement to another. We will begin with the definitions of an order-N arrangement, order-N relation composition, and a generalized homomorphism between two arrangements.

Definition 1: A simple order-N arrangement is a triple $\mathscr{A} = (F, A, D)$ where

A set of unit descriptions,
D set of possible interpretations of N unit groups based on their description,
$F \subseteq A^N \times D$ relation which gives interpretations to the relevant (ordered) groups of N units.

A general or complex arrangement is a set of simple arrangements each being defined on the same set of unit measurement descriptions and having the same label set with possibly different orders.

Definition 2: Let $F \subseteq X^N \times Z$. Let $H \subseteq X \times Y$. The order N composition of F with H is written $F \circ H$ and is defined by

$$F \circ H = \{(y_1, \cdots, y_N, z) \in Y^N \times Z \mid \text{for some } (x_1, \cdots, x_N, z) \in F, (x_n, y_n) \in H, n = 1, \cdots, N\}.$$

Order 1 relation composition is like the usual definition of relation composition except that the order in which the components are taken is slightly different. The order N composition uses the same relation H to go from each component of the x to the corresponding component of the y. Then the mth y component depends only on the mth x component, and this dependence is the same for each component.

The concept of relation composition plays a strong role in the notion of a homomorphism from one arrangement to another. A weak homomorphism is a relation which pairs or translates (nondeterministically) some of the "sense data" descriptions of the first arrangement to some of the "sense data" descriptions in the second arrangement.

After pairing or translating, the first property of the weak homomorphism is evident: the interpretive descriptions which the first arrangement gives exactly match through the homomorphism some of the interpretive descriptions which the second arrangement gives to its initial descriptions. The second property of weak homomorphisms is that they must be maximal. There cannot be any further pairings or translations included in the homomorphism without destroying the composition property.

Definition 3: Let $\mathscr{A} = (F, A, D)$ and $\mathscr{B} = (G, B, D)$ be two order-N arrangements. A *weak homomorphism* from $\mathscr{A}$ to $\mathscr{B}$ is any binary relation $H \subseteq A \times B$ satisfying:

$$F \circ H \subseteq G \tag{1}$$

$$H \subseteq H' \subseteq A \times B \text{ and } F \circ H' \subseteq G \text{ imply } H = H'. \tag{2}$$

A strong homomorphism satisfies (1) and (2) above and is also defined everywhere and single-valued; in other words, it is a function having the required composition property.

Definition 4: Let $\mathscr{A} = (F, A, D)$ and $\mathscr{B} = (G, B, D)$ be two order-N arrangements. A *strong homomorphism* from $\mathscr{A}$ to $\mathscr{B}$ is any mapping $H: A \to B$ satisfying $F \circ H \subseteq G$.

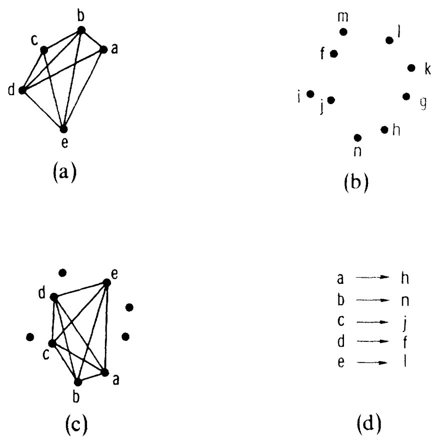

Fig. 1. (a) Given point configuration. (b) Second point configuration. The problem is to determine whether the configuration of Fig. 1(a) is contained in Fig. 2(a). (c) Shows how the point configuration of Fig. 1(a) is contained in the point configuration of Fig. 1(b). (d) Mapping from the points in Fig. 1(a) to those points in Fig. 1(b) which are a copy of the points in Fig. 1(a).

Note that a strong homomorphism has the maximality property of the weak homomorphism, since if one mapping contains another, then the two mappings must be identical.

A strong homomorphism from $\mathscr{A}$ to $\mathscr{B}$ can allow two or more elements from F to map to one element of G. Homomorphisms which are functions and one-one are called monomorphisms. Monomorphisms establish the existence of a copy of the relation F in some part of the relation G. Full isomorphisms are one-one, onto, strong homomorphisms, and they establish that the relation F is exactly like the relation G.

In the next section we illustrate a number of particular problems which are translatable to the mathematical/combinatorial problem of finding arrangement homomorphisms.

IV. Examples

A. Matching Point Configurations

The problem of matching point configurations is illustrated in Fig. 1. A set of points representing a pattern or configuration is given. The problem is to determine whether that same pattern or configuration exists in another set of points which may be scaled, rotated, reflected, or translated with respect to the first set of points. We can solve the problem by determining all the triangles in the first set of points, all the triangles in the second set of points, and then trying to match similar triangles. In the next paragraph we show how the set of triangles forms an arrangement so that the matching of similar triangles is a problem of finding a monomorphism from one arrangement to another.

Let R be an inner product space, and let D be a set of triangles, each triangle being specified by its three interior angles. We will assume that the order of the angles in the specification is not important. Let $F \subseteq R^3 \times D$ be a relation which associates with some of the triples in R^3 the name of the triangle formed by the points of the triple. Thus if R is the real plane, and the triple $(p_1, p_2, p_3) = ((0, 0), (1, 0), (1, \sqrt{3}))$ is one of the triples of R^3 to which F assigns interpretation d, then the quadruple (p_1, p_2, p_3, d) belongs to F, where d is the name for the (30°, 60°, 90°) triangle formed by the three points $(0, 0)$, $(1, 0)$, $(1, \sqrt{3})$. With these definitions it is clear that (F, R, D) is an order-3 arrangement.

Let S be another inner product space and $G \subseteq S^3 \times D$ be a relation which associates with some of the triples in S^3 the name of the triangle formed by the points of the triple. Thus (G, S, D) is also an order-3 arrangement. To determine whether there are points in S which match those in R we must determine if (G, S, D) has a copy of (F, R, D). Point matching is a problem, then, of finding a partial isomorphism from (F, R, D) to (G, S, D) (Fig. 1).

B. Image Matching

By image matching we mean how we can tell two images are of the same kind of thing. For this to happen, all the parts of one image must have similar parts in the other, and the relationships between parts in one must be the same as the relationship between the associated parts in the other.

We will illustrate how this problem can be posed as an arrangement homomorphism problem. Suppose we have a segmented image, and we are able to characterize each segment in terms of certain basic attributes, for example, shape discriminators. Using these attributes, we could assign a shape label to each of the segments. To define an arrangement from these labels, we can group related segments together, N at a time, and form the corresponding set of N-tuples of their labels.

The label given to each N-tuple can be the name we might give to a group of related segments whose shapes are the components of the given N-tuple. Another possibility is to use the interpretation label as a counter. We can assign the integer label "1" to all N-tuples arising from a group of segments the first time the N-tuple is encountered. The label k can be assigned the kth time the same kind of N-tuple is encountered.

One criterion by which segments can be considered related is spatial connectivity or nearness. Two segments are eligible to be included in the same related group when their interaction lengths overlap. To make things simple in our examples we will use interaction lengths of zero. Thus two segments are related only when they are touching.

As a specific example, one might consider a missile launching complex as described in terms of its constituent image phonemes. These might include railroad spurs, roads, power lines, buildings, radar antennas, support vehicles, etc. In terms of the stylized examples which we will present for purposes of simplicity and generality, such specific components are represented by circles, squares, triangles, etc.; however, it should be kept in mind that these "geometrical objects" are generic prototypes and always represent actual image components, shapes, attributes, subattributes, etc.

The arrangement homomorphisms can be used to establish the likeness of two images when one image is geometrically distorted from the other or when one image is essentially the same as the other, but the order or placement of the image parts is different. In either case, simple template matching will not work. The example shown in Fig. 2 illustrates one way of handling this problem using the notion of connectedness and simple order-3 arrangements. Suppose the image has four basic kinds of figures: squares, triangles, circles, and arrows. A quadruple whose first three components are these shapes taken in the order square, triangle, circle, and arrow will be considered to belong to the arrangements of the image if all three shapes touch each other in a pairwise manner. In general, we may use the criterion: consider any N-tuple if enough of its components interact in a pairwise or K-wise manner. For our example, a label of 1 or 2 will also be associated with each triple of shapes to make the quadruple; such a label will just count the number of times that the triplet with which it is associated occurs.

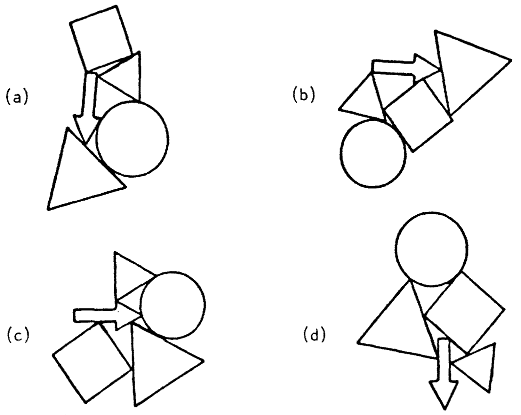

Fig. 2. Four drawings, each of which has two triangles, one square, one circle, and one arrow. Using the order-3 arrangements concept, there are two pairs of drawings whose arrangements are isomorphic.

Arrangement A	Arrangement B
(□, △, ⇧, 1)	(□, △, ⇧, 1)
(△, ○, ⇧, 1)	(□, △, ⇧, 2)
(△, ○, ⇧, 2)	(□, △, ○, 1)

Fig. 3. Quadruples in order-3 arrangements for drawings of Fig. 2. The two drawings on the left in Fig. 2 are isomorphic to Arrangement A and the two drawings on the right in Fig. 2 are isomorphic to Arrangement B. The quadruple (□, △, ↑, 2) means that the drawing has a piece that consists of a square, triangle, and arrow pairwise touching each other, and the label two designates that this is the second such piece in the drawing.

In Fig. 2, there are four drawings. Each drawing has two triangles, one circle, one square, and one arrow. Using the order-3 arrangement concept, there are two pairs of drawings whose arrangements are isomorphic by the identity function. The drawings themselves, however, have their parts placed differently in absolute position and orientation. This isomorphism becomes clear upon examination of Fig. 3, which shows the possible arrangements for the drawings. The drawings on the left are isomorphic to the arrangement labeled A. The drawings on the right are isomorphic to the arrangement labeled B.

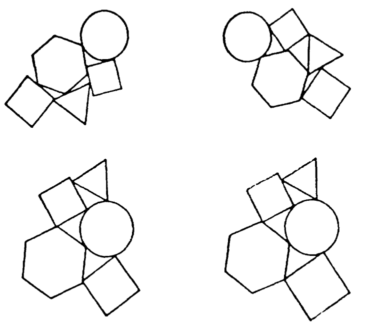

Fig. 4. Four drawings each of which has two squares, one circle, one hexagon, and one triangle. Using the order-3 arrangements concept, there are two pairs of drawings whose arrangements are isomorphic. The arrangement for each drawing is isomorphic to the arrangement for one of the drawings in Fig. 2.

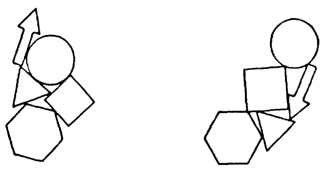

Fig. 5. Using the arrangement concept, labels of 1 or 2 can be assigned to each triplet to make one of the drawings in Fig. 2 a homomorphic image of one of these drawings.

The situation becomes slightly more complicated when the function that establishes the isomorphism is not the identity function. This is illustrated in Fig. 4 which also has four drawings. Each drawing has two squares, one circle, one hexagon, and one triangle. Taking the order as square, hexagon, triangle, and circle, and using the order-3 arrangement concept, there are two pairs of drawings in Fig. 4 whose arrangements are isomorphic. Also the arrangement for each drawing in Fig. 4 is isomorphic to the arrangement for one of the drawings in Fig. 2. The isomorphism, however, is not the identity function: a square stays square, a hexagon becomes a triangle, a triangle becomes an arrow, and a circle remains a circle.

More complicated still is the case where the correspondence between one drawing and another is by an arrangement homomorphism that does not establish a one-one correspondence. Such a case is illustrated in Fig. 5, which depicts two drawings. Taking the order as hexagon, circle, triangle, arrow, and square, and using the name or label 1 for all triplets except the triplet (arrow, triangle, square) which gets the label 2, we may use the arrangement concept to establish the correspondence between one of the drawings (the one on the right) in Fig. 5 and two of the drawings in Fig. 2 (the ones on the left). The correspondence is a homomorphism, and finding it, although easy, should begin to give the reader some idea of the combinatorial problems involved. The drawing on the left of Fig. 5 is homomorphic to neither of the drawings in Fig. 2.

The problem of finding homomorphisms is truly one of establishing the correspondence using relationships. Fig. 6 shows the quadruples in the arrangement for the right-hand drawing of Fig. 5 and the arrangement for a left-hand drawing of Fig. 2. The homomorphism which establishes the relationship between the arrangements appears in the central bottom part of Fig. 6.

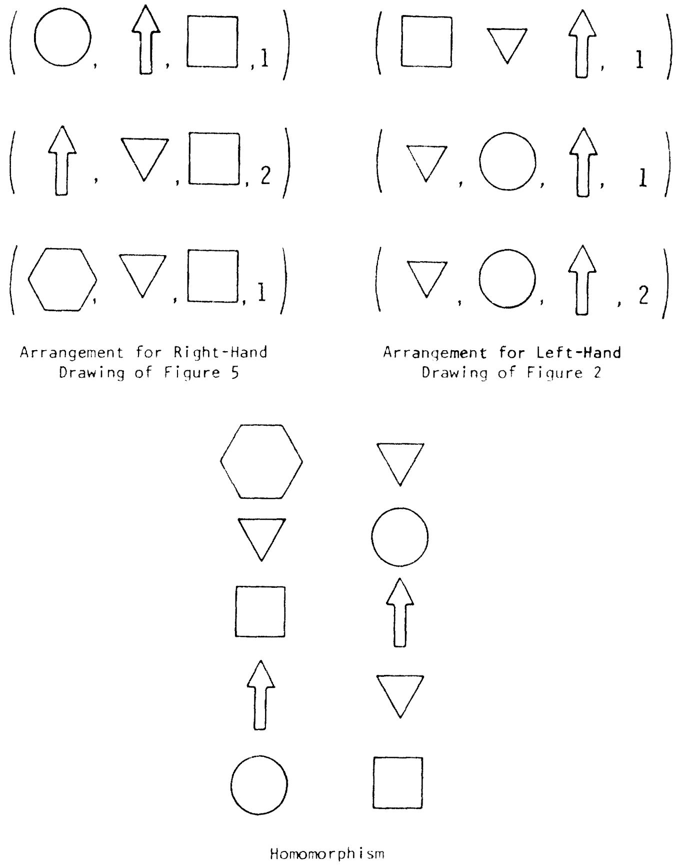
Arrangement for Right-Hand Drawing of Figure 5
Arrangement for Left-Hand Drawing of Figure 2
Homomorphism

Fig. 6. Arrangement for one of drawings in Fig. 5 and arrangement for one of drawings in Fig. 2. Below the arrangements is the homomorphism.

C. Scene Labeling

Suppose a scene has been divided into segments $S = \{s_1, \cdots, s_K\}$. A low-level feature extractor with decision rule using gray tone, color, shape, and texture of each segment assigns some possible description from a set D of descriptions to each segment. This operation defines a segment-description relation $F \subseteq S \times D$. The problem with this low-level assignment is that each segment may be associated with multiple descriptions. The desired labeling of the scene would have each segment described unambiguously.

A similar situation arises in the line labeling problem of [7]. Here, S is the set of line segments found in a scene, and D is a set containing labels that can be associated with any line. The labels in D could be, for example, convex, concave, occluding left, occluding right. The segment-description relation F, determined from low-level processes, associates with each line in S one or more labels from D. The desired line labeling would be some subset of F that associates each line with only one label.

One way of reducing the possibly ambiguous description a line or segment initially has is to use constraints from a higher level world model. Such a model can specify labeling constraints for each group of related segments or lines. To employ such a model, related (ordered) sets of N segments or lines must be determined. Segments can be related on the basis of their relative spatial positions. Lines can be related on the basis of the junctions they form. Then for each kind of relationship the model can specify a constraint which the labels of each kind of related segments or lines must satisfy.

For instance, pairs of segments in S could be related if they mutually touch each other. There could be different kinds of touching such as to the left, to the right, above, below, in front of, in back of, supported by, and contained in. Suppose L is the set of such relationship labels. Then the set of spatially related segments or lines could be specified by the relation $A \subseteq S \times S \times L$, where $(s, t, i) \in A$ if and only if label i describes the way segment s relates to segment t. In the general case, the relationships in L can describe the way N segments or lines are related so that the relation A is a labeled N-ary relation: $A \subseteq S^N \times L$.

The world model contains constraining information. For example, pairs of segments whose relationship label is i can be constrained by the world model to have associated with them only certain allowable description pairs. In this case the world model is specified as a relation $C \subseteq D \times D \times L$, where $(d_1, d_2, i) \in C$ if and only if it is legal for a pair of segments s_1 and s_2 having relation i to have respective descriptions d_1 and d_2. In general, the relation C is a labeled N-ary relation, $C \subseteq D^N \times L$ which includes in it all labeled N-tuples of compatible descriptions for an ordered set of N related segments.

To summarize the information we have available:

1) $F \subseteq S \times D$, the assignments of descriptions given by a low-level operation;
2) $A \subseteq S^N \times L$, the labeled sets of related N-tuples of segments;
3) $C \subseteq D^N \times L$, the N-ary relational labeling constraints specified by the world model.

The scene labeling problem is to use F, A, and C to determine a new labeling relation G which contains fewer ambiguous descriptions than F and which is consistent with the constraints specified by the world model. In essence we want

1) $G \subseteq F$, and
2) $A \circ G \subseteq C$.

Notice that (A, S, L) is a simple arrangement, (C, D, L) is a simple arrangement, and G is a binary relation which successfully translates the structure of arrangement (A, S, L) into the structure of arrangement (C, D, L). The binary relation G is contained in F and is a homomorphism from arrangement (A, S, L) into arrangement (C, D, L).

Note that our discussion of scene labeling is more general than that of [5], which considers only binary relational constraints. We consider N-ary relational labeling constraints; any ordered set of N segments can have a label. If we define a unique label set for each segment, then for the binary case the treatment given here exactly corresponds to that in [5].

D. Subgraph Isomorphism

Let $G = (P, E)$ and $H = (Q, F)$ be digraphs. The subgraph isomorphism problem is to determine whether there exists a subgraph of H which is isomorphic to G.

If $S \subseteq Q$, a one-one onto function $h: P \rightarrow S$ establishes the subgraph isomorphism of G to H if h satisfies $E \circ h \subseteq F$.

The refinement procedure in [9] suggested to solve this problem is a special case of the arrangement homomorphism algorithm given in the next section.

E. Spectral-Temporal Classification Using Vegetation Phenology

The usual model for classification of remotely sensed data implicitly assumes that the phenological growth stage for each vegetation category is the same for all observations made at a single time. See [3], Michigan Symposia on Remote Sensing of Environment, and Purdue Symposia on Machine Processing of Recently Sensed Data. It is well known, however, that even in a geomorphologically homogeneous area, the phenological growth stages for each vegetation type is not the same, due to differences in planting times, soil types, and weather conditions. This slop in phenological growth stage is then reflected in probability distributions of crop reflectances having larger variance than they should. The larger variance causes a lower classification accuracy for an optimal decision rule. One solution to the problem is to work from the spectral reflectance for each category to the possible phenological growth stages the category can have which are consistent with the observed spectral reflectance.

One classification algorithm which makes use of vegetation phenology has a direct and simple description. For example, if a 2-band spectral observation (α_1, α_2) is made using wavelengths (λ_1, λ_2) at time t_1, classification can be done by determining for each category c all those phenological growth stages of vegetation of category c which can yield spectral return α_1 at wavelength λ_1 and spectral returns α_2 at wavelength λ_2. If there is not a phenological growth stage of category c which yields spectral returns α_1 and α_2 at wavelengths λ_1 and λ_2, then category c is not a possible choice. At a later time t_2, if there is not a later phenological growth stage of category c which is consistent with the observed spectral reflectance, then category c is not a possible choice. Hence, classification is done by eliminating inappropriate category choices. Spectral observations taken at a later calendar time are naturally constrained to be associated with later phenological growth stages in order to keep an earlier accepted possibility of category c remaining a viable option when using observations taken at a later time.

These concepts relate to relation homomorphisms in the following way. Let B be the set of spectral bands, A be the set of measured reflectance values, G be the set of vegetation growth states, and T be a set of possible observation times. The signature for any category can be represented by a ternary relation $S \subseteq G \times (A \times B)$ which contains in it all triples (g, a, b) of (growth state, spectral reflectance value, spectral band) of high enough probability for the given category c of vegetation. The observed temporal spectral measurements for a small area ground patch which needs to be classified can also be represented as a ternary relation $W \subseteq T \times (A \times B)$ containing all triples (t, a, b) of the multitemporal observation (time, spectral reflectance value, spectral band). If there exists a monotonic function H, a homomorphism $H: T \to G$ such that $W \circ H \subseteq S$, then category c is not eliminated from consideration as the possible true vegetation category of the observed small-area ground patch. If no such homomorphism exists, the category c is not a possibility.

V. Homomorphisms

It is clear from the examples that homomorphisms for arrangements play a central role in tying together or comparing complex structures. In this section we give an algorithm for finding arrangement homomorphisms. First we will reduce the homomorphism problem for the labeled relations to a set of homomorphism problems for the unlabeled relations. Then we will analytically work on the unlabeled relation homomorphism problems showing how it can be solved by a combination of discrete relaxation and a tree search.

Suppose $F \subseteq A^N \times D$ and $G \subseteq B^N \times D$ and $H \subseteq A \times B$ satisfies $F \circ H \subseteq G$. Define $F_d \subseteq A^N$ by

$$F_d = \{(a_1, \cdots, a_N) \in A^N \mid (a_1, \cdots, a_N, d) \in F\}$$

and $G_d \subseteq B^N$ by

$$G_d = \{(b_1, \cdots, b_N) \in B^N \mid (b_1, \cdots, b_N, d) \in G\}.$$

Then clearly,

$$F \circ H \subseteq G, \qquad \text{if and only if } F_d \circ H \subseteq G_d \text{ for every } d \in D.$$

Hence the homomorphisms for the labeled relation can be determined from homomorphisms for each of the unlabeled relations.

In the remainder of this section we describe an algorithm for determining homomorphisms for unlabeled relations. The original insight into a related form of the general relaxation filtering or winnowing procedure we use is from [7] and [8]. Discussed in [5] is a more general form of the relaxation procedure in the context of the scene labeling problem with binary relational constraints. The contribution here is the development of the representation and characterization theorems for the N-ary relation homomorphisms. (See also [12].)

Let $R \subseteq A^N$ and $H \subseteq A \times B$. Recall that the composition $R \circ H$ of the N-ary relation R with binary relation H is defined by $R \circ H = \{(b_1, \cdots, b_N) \in B^N \mid \text{for some } (a_1, \cdots, a_N) \in R, (a_n, b_n) \in H, n = 1, \cdots, N\}$. Thus if each N-tuple $(a_1, \cdots, a_N)$ in R had each of its components mapped by H into the N-tuple $(b_1, \cdots, b_N)$, then the set of all N-tuples $(b_1, \cdots, b_N)$ would be the set $R \circ H$.

Let $R \subseteq A^N$ and $S \subseteq B^N$ be given. We seek to solve the equation $R \circ H \subseteq S$ for any binary relation H which is defined everywhere and single valued. Any such solution H is called a strong homomorphism of R into S. The winnowing or relaxing process plays a strong role in finding such homomorphisms, and to begin our discussion we first define our notational conventions.

Let

$$T \subseteq \mathop{\times}_{i=1}^{N} A_i.$$

We define the following sets related to T:

$$T(a_1, a_2, \cdots, a_M) = \left\{(a_{M+1}, \cdots, a_N) \in \mathop{\times}_{m=M+1}^{N} A_m \mid (a_1, \cdots, a_N) \in T\right\}$$

$$\Delta_n T = \left\{a_n \in A_n \mid \text{for some } (a_1, \cdots, a_N) \in \mathop{\times}_{i=1}^{N} A_i, (a_1, \cdots, a_N) \in T\right\}$$

$$T_n(a) = \{(a_1, \cdots, a_N) \in T \mid a_n = a\}.$$

A. The Winnowing Process

The equation $R \circ H \subseteq S$, where H is defined everywhere, says that to each N-tuple $(a_1, \cdots, a_N)$ of R, there exists at least one N-tuple $(b_1, \cdots, b_N)$ of S which is the image of the N-tuple $(a_1, \cdots, a_N)$ under the mapping H. Furthermore, the image of any N-tuple of R under H must lie in S. Thus any mapping H which satisfies the equation $R \circ H \subseteq S$ must have the following consistency property: if the element $a \in A$ is mapped to the element $b \in B$ by H, then every N-tuple of R having some component of value a can be associated, by the mapping H, with an N-tuple of S having a value of b in the corresponding component. In other words, if a mapping H purporting to satisfy $R \circ H \subseteq S$ contains the pair (a, b), and if there would exist an N-tuple of R having some component with value a, and if there were no H image of this N-tuple which is contained in S having a value b in the corresponding element, then the equation $R \circ H \subseteq S$ could not be satisfied.

Now if we begin with a given binary relation $T_1 \subseteq A \times B$ and T_1 does not satisfy $R \circ T_1 \subseteq S$, then it must be that T_1 is not consistent and has included in it too many pairs. The winnowing process is a procedure which begins with the binary relation T_1, determines which N-tuples of R can be mapped by T_1 to which N-tuples of S, and then eliminates from T_1 some of the pairs in T_1 which make T_1 inconsistent. Thus if the pair (a, b) is in T_1, and if there exists an N-tuple of R having some component with value a, and if there were no T_1-image of this N-tuple which is contained in S having a value b in the corresponding component, then the pair (a, b) is eliminated from T_1. The new relation T_2 defined by the winnowing process is, of course, contained in T_1 (Proposition 1).

Proposition 1 (Winnowing Process): Let $R \subseteq A^N$, $S \subseteq B^N$ and $T_1 \subseteq A \times B$. Assume that if for some $a \in A$, $R_n(a) = \varnothing$, $n = 1, \cdots, N$, then $\{a\} \times B \subseteq T_1$. Define $G \subseteq R \times S$ by

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \cdot S \mid (a_n, b_n) \in T_1, n = 1, \cdots, N\}.$$

Define $T_2 \subseteq A \times B$ by

$$T_2 = \left\{(a, b) \in A \times B \mid b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

Then $T_2 \subseteq T_1$.

If we let the winnowing process iterate, the successive relations it defines get smaller and smaller. Since we assume all the sets are finite, eventually the procedure converges, and we have determined a limiting relation. We should expect this limiting relation H, a "fixed point" of the winnowing process, to satisfy the equation $R \circ H \subseteq S$. And indeed, Proposition 2 states that any single-valued relation H which is a fixed point under the winnowing process must satisfy the equation $R \circ H \subseteq S$.

Proposition 2: Let $R \subseteq A^N$ and $S \subseteq B^N$. Suppose $G \subseteq R \times S$ and $H \subseteq A \times B$ satisfy

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \quad (a_n, b_n) \in H, n = 1, \cdots, N\}$$

$$H = \left\{(a, b) \in A \times B \mid b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

Then H single-valued implies $R \circ H \subseteq S$.

Thus all single-valued invariant relations under the winnowing process satisfy the equation $R \circ H \subseteq S$. But is any mapping H satisfying the equation $R \circ H \subseteq S$ a fixed point under the winnowing process? Proposition 3 states, in fact, that the winnowing process never loses a homomorphism. Actually, it proves the slightly more general result that if H is defined everywhere and satisfies $R \circ H \subseteq S$, and if $H \subseteq T_1$, then it is also true that $H \subseteq T_2 \subseteq T_1$, where T_2 is the result of one iteration of winnowing on T_1. Hence the winnowing process will reduce a relation to one which is large enough to contain all the homomorphisms it contained originally.

Proposition 3: Let $R \subseteq A^N$, $S \subseteq B^N$, $H \subseteq T \subseteq A \times B$. Define $G \subseteq R \times S$ by

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \quad (a_n, b_n) \in T, n = 1, \cdots, N\}.$$

If H is defined everywhere and $R \circ H \subseteq S$, then

$$H \subseteq \left\{(a, b) \in A \times B \quad b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

This leads to the relation homomorphism characterization theorem (Theorem 1) which states that mappings are homomorphisms if and only if they are invariant under the winnowing process.

Theorem 1. Relation Homomorphism Characterization Theorem: Let $R \subseteq A^N$ and $S \subseteq B^N$ be given. Let $H \subseteq A \times B$ be defined everywhere and single-valued. Define $G \subseteq A \times B$ by

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \quad (a_n, b_n) \in H, n = 1, \cdots, N\}.$$

Then $R \circ H \subseteq S$ if and only if

$$H = \left\{(a, b) \in A \times B \quad b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

B. Finding N-ary Relation Homomorphisms

It is clear from the characterization theorem that if the winnowing process produces a mapping for its limiting relation, then the mapping must be a homomorphism. However, the characterization theorem does not say that the winnowing process will produce relations which are either single-valued or defined everywhere. In this section we describe a representation for any homomorphism in terms of the intersections of various limiting relations produced by the winnowing process.

Suppose T^{ab} is the limiting relation determined by the winnowing process which begins with a relation whose only restriction is that the element $a \in A$ is associated with only the element $b \in B$. Then if H is a homomorphism and $(a, b) \in H$, then certainly $H \subseteq T^{ab}$. Hence, $H = \bigcap_{(a,b) \in H} T^{ab}$. Now H defined everywhere implies that $\bigcap_{(a,b) \in H} T^{ab}$ is single valued, and $\bigcap_{(a,b) \in H} T^{ab}$ single-valued and H defined everywhere imply $H = \bigcap_{(a,b) \in H} T^{ab}$.

So all homomorphisms have the representation $\bigcap_{(a,b) \in H} T^{ab}$. Is it also the case that all mappings of the form $\bigcap_{(a,b) \in H} T^{ab}$ for some defined everywhere relation H are homomorphisms? The answer is yes on the condition that the mapping $\bigcap_{(a,b) \in H} T^{ab}$ takes each N-tuple of R into some N-tuple of S. It is possible that this is not the case as illustrated in the following example. Suppose $R = \{(1, 2, 3)\}$ and $S = \{(a, b, d), (a, e, c), (f, b, c)\}$. Then the limiting relations T^{1a}, T^{2b}, T^{3c} are

$$T^{1a} = \{(1, a), (2, b), (2, e), (3, c), (3, d)\}$$

$$T^{2b} = \{(1, a), (1, f), (2, b), (3, c), (3, d)\}$$

$$T^{3c} = \{(1, a), (1, f), (2, b), (2, e), (3, c)\}.$$

Letting $H = \{(1, a), (2, b), (3, c)\}$, we find that $H = \bigcap_{(a,b) \in H} T^{ab}$, but H is not a homomorphism of R into S since it takes $(1, 2, 3)$ to (a, b, c), which is not a triple of S.

The relation homomorphism representation theorem gives the characterization that any mapping of the form $H = \bigcap_{(a,b) \in H} T^{ab}$ is a homomorphism if it takes each N-tuple of R into some N-tuple of S, and conversely, any homomorphism H has the representation $H = \bigcap_{(a,b) \in H} T^{ab}$.

Theorem 2. Relation Homomorphism Representation Theorem: Let $R \subseteq A^N$ and $S \subseteq B^N$. For each $(a, b) \in A \times B$, iteratively define the sequence of relations T_1^{ab}, T_2^{ab}, $\cdots$, T_k^{ab}, $\cdots$, by

$$T_1^{ab} = \{(a, b)\} \cup (A - \{a\}) \times B;$$

if T_k^{ab} has been defined, define T_{k+1}^{ab} by

$$T_{k+1}^{ab} = \left\{(\alpha, \beta) \in A \times B \;\middle|\; \beta \in \bigcap_{i=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_i(\alpha)} \Delta_i G^{ab}(a_1, \cdots, a_N)\right\}$$

where

$$G^{ab} = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \mid (a_n, b_n) \in T_k^{ab}, n = 1, \cdots, N\}.$$

Suppose for some integer K, $T^{ab} = T_k^{ab}$ for all $k \geq K$ and for all $(a, b) \in A \times B$. Then $H \subseteq A \times B$ defined everywhere and single valued and $R \circ H \subseteq S$ holds if and only if

1) $H = \bigcap_{(a,b) \in H} T^{ab}$ is defined everywhere and single valued, and
2) $G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \mid (a_n, b_n) \in H, n = 1, \cdots, N\}$ is defined everywhere in R.

The representation theorem allows any homomorphism to be determined by a depth-first search in the following manner. Suppose we are looking for homomorphisms which map the element $1 \in A$ to the element $a \in B$. We can determine by the winnowing process the limiting relation T^{1a} which must contain any such homomorphisms.

Now, T^{1a} may have other elements of A which are uniquely mapped to elements of B. If so, we can determine the limiting relations for these pairs and take the intersection of all of them with T^{1a}. The resulting intersection must contain any homomorphism which maps 1 to a. If the intersection has additional elements which are uniquely mapped, more intersections can be taken. When the intersection has no more additional elements which are uniquely mapped, then one of four cases exists: 1) either the intersection is not defined everywhere, in which case no homomorphism mapping 1 to a exists; 2) or the intersection is defined everywhere, is single valued, and maps each N-tuple of R into some N-tuple of S, in which case it is a homomorphism; 3) or the intersection is defined everywhere and is single valued, but cannot map some N-tuple of R into an N-tuple of S, in which case it is not a homomorphism; 4) or the intersection is defined everywhere and not single valued, in which case a choice must be made in the depth first search to map to a unique element of B one of those elements of A having possible multiple associations with elements of B. In this last case, once such a choice is made, the corresponding limiting relation must be determined and intersected with the previously intersected relations. This brings us back to the point of looking for additional uniquely mapped pairs. From here the search iterates until each branch of the tree terminates in one of the first three cases.

The actual implementation can proceed as described above or can alternatively proceed by not taking intersections but simply restricting the relation T^{1a} so that, for example, 2 is uniquely mapped to b and continuing the winnowing procedure on the restricted relation after each choice is made. In either case, it may be efficient to keep a copy of the resulting relation at each node of the tree in order to ease the computational load of the backtracking.

A = {1,2,3,4} R ⊆ A^3; R = {123,214,312,421,432,341}
B = {a,b,c,d} S ⊆ B^3; S = {abc,bad,cab,dba,dcb,cda,aaa,bbb,ccc,ddd}

	T^{1a}	T^{1b}	T^{1c}	T^{1d}
1	a	b '	c	d
2	ab	ab	c	d
3	ac	db	c	d
4	ad	cb	c	d

	T^{2a}	T^{2b}	T^{2c}	T^{2d}
1	ab	ab	c	d
2	a	b	c	d
3	da	cb	c	d
4	ca	cd	c	d

	T^{3a}	T^{3b}	T^{3c}	T^{3d}
1	a	b	ac	bd
2	a	b	bc	ad
3	a	b	c	d
4	a	b	dc	cd

	T^{4a}	T^{4b}	T^{4c}	T^{4d}
1	a	b	bc	ad
2	a	b	ac	bd
3	a	b	dc	cd
4	a	b	c	d

T^{ij} is the Limiting Relation Obtained by the Winnowing Process in which i (i ε A) is Mapped only to j (j ε B)

Fig. 7. Pair of ternary relations R and S and resulting limiting relations determined by winnowing process. It is these relations which characterize the homomorphism from R into S.

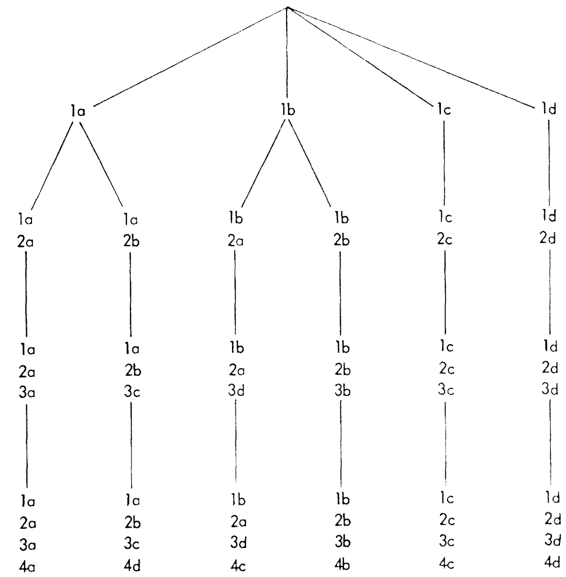

Fig. 8. Full-depth paths obtained by depth-first search which successively intersected limiting relations shown in Fig. 7. Each full-depth path is a candidate for a homomorphism from R into S.

C. Example

Fig. 7 illustrates a pair of ternary relations R and S and the resulting limiting relations determined by the winnowing process. Fig. 8 illustrates the full-depth paths obtained by the depth-first search which successively intersects the limiting relations shown in Fig. 7. Fig. 9 shows the complete search for the subtree of Fig. 8 generated by the node $1b$.

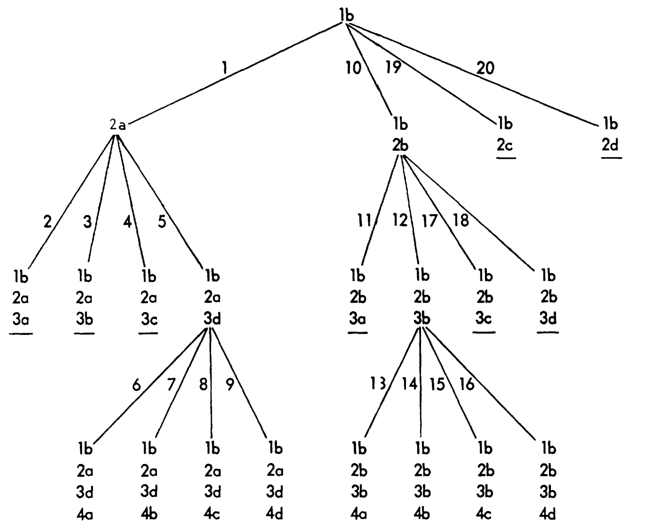

Fig. 9. Complete search of subtree generated by node 1*b*. Numbers on the branches indicate the order of the search. Paths which reach an underscore lead to inconsistencies, e.g., path 1-5-7 (1*b*, 2*a*, 3*d*, 4*b*) yields

	T^{1b} ∩	T^{2a} ∩	T^{3d} ∩	T^{4b} =	Inconsistent
1	*b*	*ab*	*bd*	*b*	*b*
2	*ab*	*a*	*ad*	*b*	∅
3	*db*	*da*	*d*	*b*	∅
4	*cd*	*ca*	*cd*	*b*	∅

Full depth (consistent) paths are candidates for homomorphisms, e.g., path 1-2-8 (1*b*, 2*a*, 3*d*, 4*c*) yields

	T^{1b} ∩	T^{2a} ∩	T^{3d} ∩	T^{4c} =	Consistent
1	*b*	*ab*	*db*	*bc*	*b*
2	*ab*	*a*	*ad*	*ac*	*a*
3	*ab*	*da*	*d*	*ed*	*d*
4	*cd*	*ca*	*cd*	*c*	*c*

VI. Complexity Analysis

Unfortunately, the arrangement homomorphism problem falls into the class of *NP*-complete problems. The complexity lies in the depth-first tree search, which if done by simple enumeration, can require in a worst case $(\#B)^{\#A} \cdot \#R \cdot \log_2 \#S$ operations, assuming S is stored in some ordered form and an operation consists of a comparison and branch. The tree search with the winnowing process added cannot guarantee any better behavior in the worst case. Fortunately, the pathological worst cases are not the ones typically encountered. For example, linear programming optimization problems are also *NP*-complete problems, yet the Simplex algorithm performs quite well for problems encountered in practice, hardly exhibiting the exponential behavior of the worst case. The Waltz filtering algorithm employed in scene labeling is usually able to reduce the tree search to just one or at most a few branches. Thus there seems to be some justification for the use of general winnowing procedures and for expecting that the resulting tree search complexity will, in the practical case, be proportional to the number of homomorphisms that exist. In the remainder of this section, we will do the complexity analysis using this kind of assumption.

Each iteration of the winnowing process takes two steps. In the first step, all N-tuples in $R \subseteq A^N$ are examined. Then for each of the N components of the N-tuple each N-tuple in $S \subseteq B^N$ must be checked to see if the value of the specified component in the S N-tuple is in the list $T(a)$, where a is the value of the specified component of the R N-tuple and $T \subseteq A \times B$. Assuming the list $T(a)$ is ordered, the number of operations this step takes is $\#R\ N\ \#S \log_2 \#B$.

In the second step, all values in the set A and all N component positions must be examined. Then all N-tuples in the relation $R \subseteq A^N$ must be located having the given value in the specified component position. Finally, intersections over a list less than $\#S$ in length must be made to determine that subset of B consistent with the original choice of the value from A. Assuming these lists are ordered, the number of operations this step takes is $\#A\ N\ \#R\ \#S\ 2\#B$. Therefore, each iteration of the winnowing process takes $N\ \#R\ \#S(\log_2 \#B + 2\#A\ \#B)$.

As mentioned at the end of Section V, there are two ways of doing the tree search. In the first way, all the limiting basis relations are calculated, and at each node in the tree the intersection of one basis relation with another binary relation needs to be done. We assume that the tree search visits no more than $\alpha K\ \#A$ nodes, where K is the number of homomorphisms, $\#A$ is the number of nodes in a complete branch, and $\alpha > 1$ is a constant indicating how much more work than the minimal amount we will have to do in the tree search. Hence, the number of operations in the tree search is $\alpha K\ \#A(2\#A\ \#B)$. Since there are $\#A\ \#B$ basis relations and the number of iterations each basis relation must participate in is no more than $\#A\ \#B$, the number of operations required to do the winnowing is $(\#A\ \#B)^2 N\ \#R\ \#S(\log_2 \#B + 2\#A\ \#B)$. Since

$$\log_2 \#B \ll 2\ \#A\ \#B,$$

the upper bound on the number of operations can be approximated by

$$2\ \#A^2\ \#B[\alpha K + N\ \#A\ \#B^2\ \#R\ \#S].$$

In the second way of doing the tree search we do not compute any basis relations and do not take intersections at nodes. Rather, at each node we restrict the relation by whatever unique value from B is going to be associated with a value from A; and then we employ the winnowing process. In this case, at least one winnowing operation must be done at each node and a maximum of $\#A\ \#B$ could be done at a node. Taking the worst case for every node yields a maximum number of operations $\alpha KN\ \#A^2\ \#B\ \#R\ \#S(2\ \#A\ \#B + \log_2 \#B)$. Since $\log_2 \#B \ll 2\ \#A\ \#B$, the upper bound on the number of operations can be approximated by $2\ \alpha KN\ \#A^3\ \#B^2\ \#R\ \#S$. For large α, this bound will be larger than the original bound. Hence, the decision on which way to do the tree search must depend on how complex the researcher thinks the tree search will be. Hard tree searches can be done the first way. Easy tree searches can be done in the second manner.

VII. Conclusions

We have discussed the mathematical construct of labeled N-ary relations which we have named order-N arrangements. We have illustrated that there are many matching problems whose abstract mathematical form is one of finding a homomorphism from one arrangement into another. We have systematically explored the structure of such homomorphisms, given a characterization and representation theorem for N-ary relation homomorphisms, and developed an algorithm for determining the homomorphisms. The algorithm consists of combining a relaxation process to find the limiting relations with a depth first tree search.

It is our hope that by illustrating 1) the underlying mathematical unity of a diverse set of problems which involves finding homomorphisms and 2) the applicability of a generalized discrete relaxation procedure to the determination of such relation homomorphisms, other more refractory problems will be able to be translated into an arrangement problem whose solution is given in this paper.

Appendix

Proposition 1. Winnowing Process: Let $R \subseteq A^N$, $S \subseteq B^N$ and $T_1 \subseteq A \times B$. Assume that if for some $a \in A$, $R_n(a) = \varnothing$, $n = 1, \cdots, N$, then $\{a\} \times B \subseteq T_1$. Define $G \subseteq R \times S$ by

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \mid (a_n, b_n) \in T_1, n = 1, \cdots, N\}.$$

Define $T_2 \subseteq A \times B$ by

$$T_2 = \left\{(a, b) \in A \times B \;\middle|\; b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

Then $T_2 \subseteq T_1$.

Proof: Let $(a, b) \in T_2$. Either $R_n(a) = \varnothing$, $n = 1, \cdots, N$ or not. If $R_n(a) = \varnothing$, then by assumption on T_1, $\{a\} \times B \subseteq T_1$ so that $(a, b) \in T_1$. If it is not the case that $R_n(a) = \varnothing$, $n = 1, \cdots, N$, then there exists an $n \in \{1, \cdots, N\}$ such that $(a_1, \cdots, a_N) \in R_n(a)$. Hence, $b \in \Delta_n G(a_1, \cdots, a_N)$ so that for some $(b_1, \cdots, b_N) \in S$, $(a_1, \cdots, a_N, b_1, \cdots, b_N) \in G$. By definition of G, $(a_1, \cdots, a_N, b_1, \cdots, b_N) \in G$ implies $(a_m, b_m) \in T_1$, $m = 1, \cdots, N$. Now $(a_1, \cdots, a_N) \in R_n(a)$ implies $a_n = a$ and $b \in \Delta_n G(a_1, \cdots, a_N)$ implies $(a_n, b) \in T_1$. Thus $(a, b) = (a_n, b) \in T_1$.

Lemma 1: Let $R \subseteq A^N$, $S \subseteq B^N$, $H_k \subseteq A \times B$, $k = 1, \cdots, K$. Define $G_k \subseteq R \times S$ by

$$G_k = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \mid (a_n, b_n) \in H_k, n = 1, \cdots, N\}.$$

If $\bigcap_{k=1}^K H_k$ is single valued, then $\bigcap_{k=1}^K G_k$ is single valued on R.

Proof: Let $(a_1, \cdots, a_N, b_1, \cdots, b_N) \in \bigcap_{k=1}^K G_k$ and $(a_1, \cdots, a_N, b'_1, \cdots, b'_N) \in \bigcap_{k=1}^K G_k$. Then for every $k = 1, \cdots, N$, we must have $(a_n, b_n) \in H_k$ and $(a_n, b'_n) \in H_k$, $n = 1, \cdots, N$. Hence, for every $n = 1, \cdots, N$, $(a_n, b_n) \in \bigcap_{k=1}^K H_k$ and $(a_n, b'_n) \in \bigcap_{k=1}^K H_k$. But $\bigcap_{k=1}^K H_k$ single valued implies that $b_n = b'_n$, $n = 1, \cdots, N$. Therefore, $\bigcap_{k=1}^K G_k$ is single valued.

Proposition 2: Let $R \subseteq A^N$ and $S \subseteq B^N$. Suppose $G \subseteq R \times S$ and $H \subseteq A \times B$ satisfy

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \mid (a_n, b_n) \in H, n = 1, \cdots, N\}$$

$$H = \left\{(a, b) \in A \times B \;\middle|\; b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

Then H single valued implies $R \circ H \subseteq S$.

Proof: Let $(b_1, \cdots, b_N) \in R \circ H$. Then for some $(a_1, \cdots, a_N) \in R$, $(a_n, b_n) \in H$, $n = 1, \cdots, N$. By definition of H,

$$b_k \in \bigcap_{n=1}^{N} \bigcap_{(a_1', \cdots, a_N') \in R_n(a_k)} \Delta_n G(a'_1, \cdots, a'_N), \qquad k = 1, \cdots, N.$$

But $(a_1, \cdots, a_N) \in R_k(a_k)$ so that $b_k \in \Delta_k G(a_1, \cdots, a_N)$, $k = 1, \cdots, N$. Now, by Lemma 1, H single valued implies G is single valued. And $b_k \in \Delta_k G(a_1, \cdots, a_N)$ certainly implies $G(a_1, \cdots, a_N) \neq \varnothing$. Hence, there exists a unique $\{(b'_1, \cdots, b'_N)\} = G(a_1, \cdots, a_N)$. Then, $b_k \in \Delta_k G(a_1, \cdots, a_N)$, $k = 1, \cdots, N$, and $\{(b'_1, \cdots, b'_N)\} = G(a_1, \cdots, a_N)$ imply $b_k = b'_k$, $k = 1, \cdots, N$ so that $(b_1, \cdots, b_N) \in G(a_1, \cdots, a_N) \subseteq S$.

Lemma 2: Let $R \subseteq A^N$, $S \subseteq B^N$, $H \subseteq T \subseteq A \times B$. Define $G \subseteq R \times S$ by

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \mid (a_n, b_n) \in T, n = 1, \cdots, N\}.$$

Then $R \circ H \subseteq S$, $(a_1, \cdots, a_N) \in R$, and $(a_m, b_m) \in H$, $m = 1, \cdots, N$ imply $b_k \in \Delta_k G(a_1, \cdots, a_N)$, $k = 1, \cdots, N$.

Proof: Let $(a_1, \cdots, a_N) \in R$ and $(a_m, b_m) \in H$, $m = 1, \cdots, N$. Since $R \circ H \subseteq S$, we must have $(b_1, \cdots, b_N) \in S$. Since $H \subseteq T$, $(a_m, b_m) \in H$ implies $(a_m, b_m) \in T$, $m = 1, \cdots, N$. Now, $(a_1, \cdots, a_N) \in R$, $(b_1, \cdots, b_N) \in S$, and $(a_n, b_n) \in T$, $n = 1, \cdots, N$ imply $(a_1, \cdots, a_N, b_1, \cdots, b_N) \in G$. Hence, $(b_1, \cdots, b_N) \in G(a_1, \cdots, a_N)$ so that $b_k \in \Delta_k G(a_1, \cdots, a_N)$, $k = 1, \cdots, N$.

Proposition 3: Let $R \subseteq A^N$, $S \subseteq B^N$, $H \subseteq T \subseteq A \times B$. Define $G \subseteq R \times S$ by

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \mid (a_n, b_n) \in T, n = 1, \cdots, N\}.$$

If H is defined everywhere and $R \circ H \subseteq S$, then

$$H \subseteq \left\{(a, b) \in A \times B \;\middle|\; b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

Proof: Let $(a, b) \in H$. We need to show that

$$b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N).$$

If $R_n(a) = \varnothing$, $n = 1, \cdots, N$, the required result is immediate. If $R_n(a) \neq \varnothing$ for some n, then let $n \in \{1, \cdots, N\}$ and $(a_1, \cdots, a_N) \in R_n(a)$ be given. Since $(a_1, \cdots, a_N) \in R_n(a)$, $a_n = a$. For convenience we set $b_n = b$. Since H is defined everywhere, there exists $b_1, \cdots, b_{n-1}, b_{n+1}, \cdots, b_N \in B$ such that $(a_k, b_k) \in H$, $k \neq n$. Now by Lemma 2, $R \circ H \subseteq S$, $(a_1, \cdots, a_N) \in R$, $(a_m, b_m) \in H$, $m = 1, \cdots, N$ imply $b_i \in \Delta_i G(a_1, \cdots, a_N)$, $i = 1, \cdots, N$. In particular then, $b_n \in \Delta_n G(a_1, \cdots, a_N)$. Therefore,

$$b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N).$$

Theorem 1. Relation Homomorphism Characterization Theorem: Let $R \subseteq A^N$ and $S \subseteq B^N$ be given. Let $H \subseteq A \times B$ be defined everywhere and single valued. Define $G \subseteq A \times B$ by

$$G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \quad (a_n, b_n) \in H, n = 1, \cdots, N\}.$$

Then $R \circ H \subseteq S$ if and only if

$$H = \left\{(a, b) \in A \times B \,|\, b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

Proof: Suppose $R \circ H \subseteq S$. Since H is defined everywhere, by Proposition 3,

$$H \subseteq \left\{(a, b) \in A \times B \quad b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\} = F.$$

Now suppose $(a, b) \in F$. Then

$$b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N).$$

Let $n \in \{1, \cdots, N\}$ and $(a_1, \cdots, a_N) \in R_n(a)$. Then $a_n = a$ and $b \in \Delta_n G(a_1, \cdots, a_N)$. By definition of G, $b \in \Delta n G(a_1, \cdots, a_N)$ implies $(a_n, b) \in H$. But $(a, b) = (a_n, b) \in H$. Hence $F \subseteq H$. $F \supseteq H$ and $F \subseteq H$ imply $F = H$. Suppose

$$H = \left\{(a, b) \in A \times B \quad b \in \bigcap_{n=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_n(a)} \Delta_n G(a_1, \cdots, a_N)\right\}.$$

Since H is single valued, by Proposition 2, $R \circ H \subseteq S$.

Lemma 3: Let $F \subseteq A \times B$ and $G \subseteq A \times B$. Suppose F is defined everywhere and G is single valued. The $F \subseteq G$ implies $F = G$.

Proof: Suppose $F \subseteq G$. Let $(a, b) \in G$. Since F is defined everywhere, there exists a $b' \in B$ such that $(a, b') \in F$. But $F \subseteq G$ and $(a, b') \in F$ imply $(a, b') \in G$. But because G is single valued, $(a, b) \in G$ and $(a, b') \in G$ imply $b = b'$. Hence $(a, b) = (a, b') \in F$ so that $G \subseteq F$. Finally, $F \subseteq G$ and $G \subseteq F$ imply $F = G$.

Theorem 2. Relation Homomorphism Representation Theorem: Let $R \subseteq A^N$ and $S \subseteq B^N$. For each $(a, b) \in A \times B$, iteratively define the sequence of relations $T_1^{ab}, T_2^{ab}, \cdots, T_k^{ab}, \cdots$, by

$$T_1^{ab} = \{(a, b)\} \cup (A - \{a\}) \times B;$$

if T_k^{ab} has been defined, define T_{k+1}^{ab} by

$$T_{k+1}^{ab} = \left\{(\alpha, \beta) \in A \times B \quad \beta \in \bigcap_{i=1}^{N} \bigcap_{(a_1, \cdots, a_N) \in R_i(\alpha)} \Delta_i G^{ab}(a_1, \cdots, a_N)\right\}$$

where

$$G^{ab} = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \quad (a_n, b_n) \in T_k^{ab}, n = 1, \cdots, N\}.$$

Suppose for some integer K, $T^{ab} = T_k^{ab}$, for all $k \geq K$ and for all $(a, b) \in A \times B$. Then $H \subseteq A \times B$ defined everywhere and single valued and $R \circ H \subseteq S$ holds if and only if

1) $H = \bigcap_{(a,b) \in H} T^{ab}$ is defined everywhere and single valued, and
2) $G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) R \times S \,|\, (a_n, b_n) \in H, n = 1, \cdots, N\}$ is defined everywhere in R.

Proof: Suppose $R \circ H \subseteq S$ and H is defined everywhere and single valued. By Proposition 3, $H \subseteq T^{ab}$ for each $(a, b) \in H$. Hence $H \subseteq \bigcap_{(a,b) \in H} T^{ab}$. Since $\{b\} = T^{ab}(a)$ for each $(a, b) \in H$, and H is defined everywhere, $\bigcap_{(a,b) \in H} T^{ab}$ is single valued. Now by Lemma 3, $H \subseteq \bigcap_{(a,b) \in H} T^{ab}$ with H defined everywhere and $\bigcap_{(a,b) \in H} T^{ab}$ single valued, imply $H = \bigcap_{(a,b) \in H} T^{ab}$. Let $G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \,|\, (a_n, b_n) \in H, n = 1, \cdots, N\}$ and suppose $(a_1, \cdots, a_N) \in R$. Since H is defined everywhere, there exists $(b_1, \cdots, b_N) \in B^N$ such that $(a_n, b_n) \in H$, $n = 1, \cdots, N$. Since $R \circ H \subseteq S$, $(a_1, \cdots, a_N') \in R$ and $(a_n, b_n) \in H, n = 1, \cdots, N$ imply $(b_1, \cdots, b_N) \in S$. Hence, $(a_1, \cdots, a_N, b_1, \cdots, b_N) \in G$ so that G is defined everywhere in R. Suppose $H = \bigcap_{(a,b) \in H} T^{ab}$. Suppose that $G = \{(a_1, \cdots, a_N, b_1, \cdots, b_N) \in R \times S \,|\, (a_n, b_n) \in H, n = 1, \cdots, N\}$ is defined everywhere in R. If $(b_1, \cdots, b_N) \in R \circ H$ we want to show $(b_1, \cdots, b_N) \in S$. Let $(b_1, \cdots, b_N) \in R \circ H$. Then there exists $(a_1, \cdots, a_N) \in R$ such that $(a_n, b_n) \in H$, $n = 1, \cdots, N$. Since G is defined everywhere in R, there exists some $(b_1', \cdots, b_N') \in S$ such that $(a_1, \cdots, a_N, b_1', \cdots, b_N') \in G$. But $(a_1, \cdots, a_N, b_1', \cdots, b_N') \in G$ implies $(a_n, b_n') \in H$, $n = 1, \cdots, N$. Now $(a_n, b_n) \in H$ and $(a_n, b_n') \in H$ imply $b_n = b_n'$ since H is single valued. Hence $(b_1, \cdots, b_N) = (b_1', \cdots, b_N') \in G(a_1, \cdots, a_N) \subseteq S$.

REFERENCES

[1] H. G. Barrow, A. P. Ambler, and R. M. Burstall, "Some techniques for recognizing structures in pictures," in *Frontiers in Pattern Recognition*, Watanabe, Ed. New York: Academic, 1972, pp. 1-29.

[2] E. F. Codd, "A relational model for large shared data banks," *Commun. Assoc. Comput. Mach.*, vol. 13, p. 377, 1970.

[3] K. S. Fu, D. A. Landgrebe, and T. L. Phillips, "Information processing of remotely sensed agricultural data," in *Proc. IEEE*, vol. 57, no. 4, pp. 639-653, Apr. 1969.

[4] F. Hayes-Roth, "Representation of structured events and efficient procedures for their recognition," *Pattern Recognition*, vol. 8, no. 3, pp. 141-150, July 1976.

[5] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by

relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 6, pp. 420–433, June 1976.

[6] H. G. Barrow and J. M. Tenenbaum, "MYSYS: A system for reasoning about scenes," SRI Artificial Intelligence Center, Tech. Rep. No. 121, Stanford Research Institute, Menlo Park, CA, 1976.

[7] D. L. Waltz, "Generating semantic descriptions from drawings of scenes with shadows," MIT Tech. Rep. A1271, Nov. 1972.

[8] ——, "Automata theoretic approach to visual processing," in *Applied Computation Theory*, R. T. Yeh, Ed. Englewood Cliffs, NJ, Prentice-Hall, 1975.

[9] J. R. Ullman, "An algorithm for subgraph isomorphism," *J. Assoc. Comput. Mach.*, vol. 23, no. 1, pp. 31–42, Jan. 1976.

[10] M. Minsky, "A framework for representing knowledge," in *The Psychology of Computer Vision*, P. Winston, Ed. New York: McGraw-Hill, 1975.

[11] A. Hanson and E. Riseman, "A progress report on vision: Representation and control in the construction of visual models," COINS TR 76-9, Univ. Massachusetts, Amherst, July 1976.

[12] R. M. Haralick, "The characterization of binary relation homomorphisms," *Int. J. General Syst.*, vol. 4, pp. 113–121, 1978.

Embedded Game Analysis of a Problem in International Relations

JOHN V. GILLESPIE, DINA A. ZINNES, AND GURCHARAN S. TAHIM

Abstract—**The behavior of two groups of nations, superpowers and regional powers, engaged in an arms race represented as a differential game are studied. The various nations are assumed to arm themselves according to a balance of power, the policemen, or the second attack rationale. The regional nations are assumed to control the disposition of strategic resources desired by the superpowers. To obtain these, the superpowers provide assistance to the regional nations. The optimal trajectory corresponding to the above three rationales is computed, and the stability of the trajectory is studied. After a change in the international situation, the regional nations may alter the distribution of the material to the various nations. The effect of this on the stability of the resulting optimal trajectory is investigated.**

I. Introduction

THE MAJOR AIM of this analysis is to study, in the spirit of Richardson's pioneering work [26], the behavior of two groups of nations in an arms race. More recently the notions of differential games have been applied successfully to the study of the behavior of nations involved in arms races [8], [16], [17], [29]. An important part of the above studies is that they relate to the investigation of armament behavior from the perspective of a single nation. Of course, the dynamics of the arms race imply that the behavior of the single nation is influenced by the other nations in the race. However, these studies do not examine simultaneously the performance of other nations engaged in the race.

This paper can be viewed as the next step in modeling arms races.[1] In this study there are two groups of nations involved in a dynamic interaction represented by a differential game. Each group consists of two nations: the first group represents stronger nations or coalitions of nations that are labeled as *superpowers* (e.g., United States and Soviet Union), while the second group represents less powerful nations or coalitions labeled as *regional powers* (e.g., Israel and the Arab States). The four nations or coalitions are involved in a differential game. We will give the performance indices representing rationales commonly offered in the descriptive international relations literature.

A. The Simple Embedded Game

The superpowers are engaged in a Nash game [9]; the regional powers are also engaged in a Nash game among themselves. Each superpower provides assistance to a corresponding regional power. The two strategies for providing assistance from the superpowers to the regional actors constitute a Nash equilibrium pair. The superpowers game, the regional powers game, and the assistance game, will all be referred to as an embedded game [16]. To study the optimal behavior of the nations in the embedded game, the various performance indices we will consider are balance of power, second attack, and policemen. These three performance indices are drawn from the rationales for defense policy commonly suggested by scholars of international affairs.

Manuscript received August 29, 1977; revised March 22, 1978. This work was supported in part by the National Science Foundation under Grant SOC 75-04212, in part by the Ford Foundation under Grant 750-0514.

The authors are with the Center for International Policy Studies, Department of Political Science, Indiana University, Bloomington, IN 47401.

[1] Our analysis should be viewed as an application of known systems techniques to a problem in international relations. Our purpose is not to contribute basic ideas about systems theory but to indicate a new and viable area of its application.

0018-9472/78/0800-0612$00.75 © 1978 IEEE