COMPUTER PERCEPTION OF COMPLEX PATTERNS

Edward L. Morofsky and Andrew K. C. Wong
Biotechnology Program
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

A software Pattern Perception System, PPS, motivated by human perceptive characteristics, is developed to recognize and classify complex line patterns. This paper presents the functional organization of PPS, as well as relevant psychological observations. The data structure and processing methodology involved are also illustrated with a sample pattern. Recursive patterns are treated, specifically their representation and generation.

Introduction

A software pattern Perception System, PPS, has been recently developed for recognizing complex line patterns by incorporating some of the perceptive characteristics observed in perceptual psychology. This approach to structured pattern recognition is introduced as an alternative to the contemporary syntax-directed analysis approaches (1)(2)(3), with the objective of (a) increasing the flexibility and efficiency of the system, and (b) experimenting with various ways of enhancing the visual perceptive capability of computers. The purpose of this paper is to establish the psychological relevance of PPS and to describe the functional organization of the system. Examples of PPS activities related to recognition and class formation will also be presented and discussed in this paper. The detailed formal information structures and processing facilities, however, will be reported In another paper by these authors (4).

At present, PPS is implemented on the APL/360 on-line system utilizing extensively the distributive programming environment available in that language. While the concept of PPS Is not confined by the programming language, we should point out that many of the numerical codes that were created represent a special symbolism based on linear lists and arrays.

PPS and Perceptual Psychology

Seeing a pattern implies its isolation as a figural unit from the rest of the visual field, while perceiving it requires, in addition, the assignment of an organized representation to the pattern (5). This organized representation is referred to as the perceived pattern and is a product of the constructive aspect of perception. In PPS the analogous representation of a pattern consists of both a structured description which groups the pattern components into an hierarchy and a local description of the components in terms of a limited set of primitive forms.

Psychological experiments have shown that different subjects, when asked to memorize meaningless patterns and reproduce them later in terms of perceptual units, decomposed patterns into various sets of components which were dependent on the strategies used by the observers (6). Figure 1 illustrates three ways in which the pattern given in Fig. Ia can be decomposed. We call the strategy associated with the decomposition in Fig. Ib the continuity strategy, since each component is traced until it terminates at a free end or at a junction with itself or another line. Figure Ic exhibits another strategy which extracts the closed perimeter or outer boundary - a line which encloses the entire pattern - and fills in the remaining components. A third strategy operates by tracing around the simplest enclosed areas comprising the pattern. The components thus obtained are shown in Fig. 1d, which differs from Fig. Ib and Ic in that it represents planar regions rather than lines. Thus, in both PPS and the human perceptive process, a segmentation of flgural units is directed by some preconceived rules, as well as by the pattern features.

It has been recognized (5)(7) that components or objects segregated after the preattentive processes provide the potential framework for subsequent pattern synthesis and analysis. In general, the manner in which a pattern is segmented determines the reorganization of the components into an hierarchical description and further affects the pattern analysis. The way in which components are organized is of fundamental concern in cognitive psychology and also in the implementation of PPS. Figure 2a shows a complex pattern with components A, B, C, D and E. Figure 2b and 2c show two different hierarchical representations according to the choice of reference component with which the other components are grouped. Here, the representation of Fig.2b is intuitively judged to be simpler since it requires only two levels. However, when two hierarchical representations of the same pattern (Fig. 3) have an equal number of hierarchical levels, a further discriminating criterion is necessary to select the one with greater organizational simplicity. At present, there is no objective measure of organizational simplicity (8). A method is needed to express and extend the intuitive judgment used heretofore. The measure incorporated in PPS, which has been found to be consistent with intuition and within the system, is one in which the structure exhibiting the lowest average hierarchical level is judged to be the simplest. This measure for a given hierarchy can be represented by L_{av}, the average hierarchical level,

$$L_{av} = \frac{\sum_{i=1}^{N} n_i \cdot i}{\sum_{i=1}^{N} n_i} \quad \ldots\ldots\ldots\ldots\ldots [1]$$

where n_1 is the number of components at level 1 and N is the order of the greatest level. Figure 3b has one first-level, three second-level, two third-level and one fourth-level component, yielding an average component hierarchical level of 2.43 versus that of 2.57 obtained from the representation of Fig. 3c. Therefore, the structure with component A as reference is preferred over that with component B, agreeing with common intuition. The recommended measure, termed the simplicity criterion in FPS, tends to choose a structure which groups the most components at levels near the reference component. For structures with equal simplicity, local descriptions are required in the selection of the reference.

The simplicity criterion may sometimes be superseded, particularly when the pattern contains a meaningful subpattera (6), i.e., one that is recognized by the observer. In such cases, a familiar subpattera or a subpattern with distinctive features may be extracted as a unit and used to group the remainder of the pattern. Figure 4a is a pattern with components A, B, C and D found from the continuity strategy and Fig. 4b shows the hierarchical structure with A as the reference. If the observer sees components C and D as the character "4", then the structure of Fig. 4c is the likely representation, even though it has more hierarchical levels. Of course, no pattern is completely meaningless, so that a continual tension is maintained between organizing a pattern according to the simplicity criterion and organizing around a more or less familiar subpattern.

The existence of hierarchical representations in human perception is supported by retinal stabilization studies (9)(10), which indicate that a pattern disappears and reappears in fragments that are closely grouped in a hierarchy. Meaningful or recognizable patterns tend to remain intact more than unfamiliar ones, indicating the influence of previous experience on perception.

Eye movement studies (11) have indicated that the preliminary observation of a painting or pattern is an essentially non-directed survey of the whole, during which an initial representation is constructed. Focal attention is then brought to bear for a more detailed examination of interesting features. If non-directed processing is assumed, an objective measure for reorganizing the initial representation is required. Different observers or the same observer at different times may form various schemata of the same pattern (6). The formation of a schema depends on the choice of reference, as well as the structure into which component sets are grouped. The susceptibility to variation in the structured description (due to indecision as to the choice of reference) may be viewed as an indication of the instability of the organization of the perceived pattern. A pattern that can be represented by hierarchical structures of equal stability or simplicity is considered to be an ambiguous pattern. The perceived pattern may then shift from one representation to another. Figure 5a is a pattern that has been decomposed into two sets of two components each, shown in Fig. 5b and Fig. 5c, by the continuity and outer boundary heuristics, respectively. The two representations have equal simplicity, although the decomposition into two triangles has more familiar components and is thus more frequently seen as such. Therefore, In PPS, unambiguous patterns can be uniquely represented in an hierarchical structure of maximum simplicity, while ambiguous patterns may have more than one hierarchical structure of equal simplicity. Once a representation of maximum simplicity is achieved and the corresponding L obtained is much less than those of other representations, PPS will retain that as a stable representation unless it is redirected, by command, to consider another component as a new reference. This process may resemble the phenomenon of fixation in which one representation completely eliminates a plausible alternative representation, except for deliberate efforts by the perceiver to reconstruct the pattern.

After the pattern is synthesized into an organizational whole, detailed processing of local information proceeds. PPS divides the lines into segments of various curvatures and records junctions and angles at the same time. This processing is compatible with the results of experimental studies of the information content of lines and the manner in which they are processed. Hochberg (12) found that the eye fixates at points of discontinuity and at angles, while Attneave (13) showed that the information content of a contour is greatest at points where the contour deviates from a straight line. Thus, in PPS all line components are expressed in the form of a code sequence of curve segments, angles and junctions. These components, together with their overall interconnected topological structure, constitute a complete description of a pattern.

Another important aspect of the greater flexibility and efficiency achieved by PPS in modelling perception is its information storage and retrieval processes. The retrieval and recognition processes of PPS, in line with several respected hypotheses of visual memory (5)(14), consist of two stages - primary and secondary. The former is a crude, whollstlc and parallel "primary process," whereas the latter is an elaborate "secondary process" that includes deliberate manipulation of information by an active agent. Thus, remembering is not only a process of arousing "memory traces," but also a constructive process making use of the stored Information to build something new - a method similar to the combination of preattentive and "analysls-by-synthesis" processes noted in human perception (5).

In PPS the information structure of the pattern universe (PU), containing the memory storage of pattern classes, and the organized description of a scene (D), are represented by a heterogeneous array with well-defined information fields. By using hash code techniques, detailed information for each component in both arrays can be suppressed and reformed when necessary, whereas the joint occurrence of specific qualitative features (or partial characteristics) in each line component can be represented by a product of prime numbers known as the feature code and stored explicitly in both arrays. When a basic form has been segregated in D, with its feature characteristics extracted from the feature code, a parallel search process on the PU is performed to reduce the search space to a subset of PU containing those pattern classes which share similar features with D in the basic form of their reference components. This search process, which operates on the feature code, is efficient, parallel and exhaustive, resembling the primary processing of image recall. Besides this retrieval process of search for similarity, the patterns or subset of patterns stored can be called by a specific name or a class name. Regardless of the manner in which a subset of PU is called, the topological structure of D Is subsequently compared with each pattern cell of the subset sequentially, level by level, until a small subset of PU is retained for detailed analysis and comparison. Then, further details of components can be recalled and reformed by the hash code technique. A constructive process is then underway. Furthermore, if a certain subpattern of a scene is identified during the search, its relation to the other recognizable patterns or meaningless subpatterns can also be deduced. Thus in PPS, identifying, interpreting, or classifying a pattern (or a group of patterns) involves

(a) a crude parallel search of the memory,

(b) a sequential search for structural similarity, and

(c) a detailed synthesis and analysis process after some images have been recalled. (The synthesis or constructive process is more obvious when the objects being identified belong to the recursive pattern class.)

Functional Organization of PPS

The functional organization of PPS is shown schematically in Fig. 6. Here the major processing units of the system are represented by rounded blocks, while the external and internal data are represented by square blocks. Rules or criteria, external to the system, are enclosed in square brackets and some of the psychological counterparts are listed in quotations.

The Function TRACE

PPS accepts patterns that have been transformed into a matrix representation, (Fig. 7), known as a field matrix, of l's and 0's representing, respectively, the presence and absence of line segments inside a mesh element. Both the triangular[+] and rectangular meshes are employed in PPS in the representation of a digitized picture. A function called TRACE traces along line segments according to several available rules. In PPS, the starting point of the trace process is arbitrary,* that is, no global syntax is required in the extraction of information from the field matrix. Components of the pattern are then encoded into a code string with junctions labelled and recorded. Thus, code strings are first recorded in terms of bearing segments (using the eight bearings to the eight immediately adjacent neighboring elements) and junction labels are later reduced into sequences of curvature, bearing and size triplets. These component strings, together with a tentative structural description, give the complete arbitrary description of the scene.

As an illustrative example, Fig. 7 is a complex pattern representing an apple outline interlinked to an enclosed character "3" shown with its associated field matrix. The field matrix is then processed and in this case the tip of the stem is chosen as the starting point. During the trace process, the function TRACE assigns separate labels to each junction, (VIZ. -2, -3, -A, -5, -6 in Fig. 8) according to the order of trace. In addition to labelling, TRACE also records the junction type for each junction. Here, junctions -2, -3, -A, -5 and -6 are T-junctions, whereas -7 is an intersection. While tracing is in process, TRACE also records the bearing of each segment and encodes both the junction and bearing information in sequential codes. Figure 10 shows the code strings for all components. In terms of junction label and segment bearings, the code string of the apple stem is

- 2 1 1 1 2

signifying that it has three segments in direction 1, one in direction 2 and terminates at the T-junction labelled as -2. This is then reduced to curvature, bearing and size triplets.

[+] A triangular scheme is under development for PPS which provides more segment orientation and makes the regeneration of the pattern from D more convenient.

*Except that the orientation and position should be recorded if the orientation and position of the picture are desired.

In this example, one may note that the stem of the apple is chosen as the arbitrary reference and occupies level one of the tentative hierarchy The apple outline thus assumes the second level as it has a junction (-2) in common with the stem. There are, in all, two third-level components and one fourth-level component. Therefore, in this arbitrary representation the following label notation is assumed

1 1 - reference component - stem
2 1 - second level component - apple outline
3 1 - third level component - T-junction with "3"
3 2 - third level component - Intersects with "3"
4 1 - fourth level component - character "3"

and the arbitrary topological descriptive string (ATDS) is as follows:

$$1\ 1 \overset{\circ}{\dashv} (2\ 1 \overset{I}{\vdash} (3\ 1 \dashv 4\ 1) \overset{I}{\models} (3\ 2\ X\ 4\ 1)).\ .[2]$$

with

$\dashv$, $\vdash$: T-junctions.
X : Intersection of two components.
$=\!\!\mid$, $\models$: Double T-junctions of one component attaching to the other.
I : Terminating component contained within a closed component.
° : Terminating component not contained within a closed component.

In the ATDS, each component Is first listed and followed by pairs of < relation, component >. Thus, the component (11) has one pair and component (2 1) has two relational pairs with components (3 1) and (3 2), respectively.

Hence,

$$A \overset{\circ}{\vdash} B$$

signifies that A attaches to B at a T-junction, A being continuous and B terminating with B not enclosed in A. On the other hand,

$$B \overset{I}{=\!\!\mid} A$$

implies B is connected to A in a double T-junction with A continuous and B terminating, with B enclosed by the closed component A. With this tentative topological hierarchy, the ATDS with all its variables defined (i.e., (1 1), (2 1), etc.) contains all information required for the full description of the pattern.

The Function REORGANIZE

After the complete description of a pattern is stored in the system, PPS attempts to simulate the second level of the perceptive activities--the reorganization of the image into a meaningful pattern according to various organizational criteria. As mentioned in the previous sections, one of the criteria PPS uses is organizational simplicity. With this criterion the unambiguous reference chosen is the apple outline, which is also the only closed component and has the greatest length and number of junctions. Once a reference component is chosen, the function REORGANIZE is able to transform the ATDS into a specific topological descriptive string, STDS, which is a specific topological description of the pattern.

In the previous example, the choice of a new reference would lead to a reorganization of the hierarchy as indicated by the following reassignment of hierarchical identifiers:

1 1 - apple outline
2 1 - intersecting arc
2 2 - stem
2 3 - connector between "3" and apple outline
3 1 - the number "3"

and the STDS would be

$$1\ 1 \overset{I}{\models} (2\ 1\ X\ 3\ 1) \overset{\circ}{\vdash} (2\ 2) \overset{I}{\vdash} (2\ 3 \dashv 3\ 1).[3]$$

It should be noted that the hierarchical structure of the ATDS .contains four levels, while the STDS contains three. If we apply the simplicity criterion the ATDS yields an average hierarchical level of 2.6, while STDS has an average hierarchical level of 2.0. Thus, as it should, the STDS displays a greater simplicity as measured by the simplicity criterion.

Although STDS contains the complete description of the pattern, it is transformed into a compact array for more efficient processing. This array, with well-defined information fields for each column, is called the description, D, of the pattern. The D resulting from STDS in Eq. [3] and component information from Fig. 7, is given in Fig. 10. In D, columns 1 and 2 comprise the referee label signifying the hierarchical level and Index of the component respectively. In like manner, Columns 3 and 4 together are called the referent label, indicating to which component the referee is connected. Columns 1 through A are termed the linkage field as they list which components are linked. In D, special features of each component (such as "open-endedness," "self-intersection," etc.) are extracted and stored explicitly in terms of the product of prime numbers, each of the prime numbers represents a distinct component feature. This crude component feature information, abstracted from each component string, is represented in a feature code (FC). A similar junction code (JC) describes the type of junction and the relations between the junction components

FC and JC are stored in columns 5 and 6 of D respectively. These codes will later be used to direct the extraction of a sub-universe of PU to reduce the search space in the recognition phase. In D, the detailed information of each component is not explicitly stored, but its corresponding hash code is stored in column 7 of D. The detailed information is dumped to a "character pool" (a string of characters) and is recoverable (reconstructed) when detailed comparison of components is required.

Pattern Class Generation

If an object pattern is distinct, unambiguous and unique, such as a "circle," "square" or "apple," its representation in D can be transferred to the permanent memory storage to form the concept of square, circle or apple. In PPS, this permanent memory storage is known as the pattern universe, PU. The generation of a cell in the PU containing information which defines a class pattern is referred to as pattern class generation. Cells in the PU are modifiable. New cells can be added and old cells can be modified by adaptive learning or teaching by reinforcement of local line forms or of the topological structure of the class pattern.

A class need not necessarily be named. If the pattern class of a PU cell is named, the hash code of the name will be stored in the name field (Fig. 11) and the cell can be recalled by name. If no name has been assigned, the cell can only be extracted by specific features existing in its reference or other components, or by similarity of topological structure.

In general, the D of a stable pattern of fixed hierarchical structure is organizationally similar to its corresponding PU cell, except that the PU cell contains three additional column fields. They are: (a) class name field, (b) component status field, and (c) weighting factor field. In the PU cell, the component status field specifies the essential and non-essential components comprising a pattern class definition. At the present stage, three types of status can be assigned in the component status field. They are defined as follows:

Let S_{ij} represent the status of a component C_{ij}. Then

$$S_{ij} = \begin{cases} 1, & \text{if } C_{ij} \text{ is essential to the class;} \\ 0, & \text{if } C_{ij} \text{ is non-essential, but not alien to the class;} \\ -N, & \text{if there exists at least } N\ C_{ij} \text{ at level } i \text{ to satisfy the class definition.} \end{cases}$$

Hence, in "cell-apple" of the PU in Fig. 11, the apple outline is essential and the stem is non-essential, but is not considered to be an object external to the pattern. This concept is important in discerning an object embedded in a scene. Any additional part, attached to an identifiable object and not corresponding to a non-essential component at its connection level, is considered to be external to the pattern. Hence, PPS is capable of determining how an identifiable pattern is related to another.

The weighting field in the PU cell contains the weighting factor, W_{ij}. During adaptive learning it updates the ratio of the number of component occurrences to the number of similar patterns submitted. Hence, in PPS, the process of adaptive learning simply implies a continual matching of a set of given D's with their corresponding PU cell and a simultaneous modification of the component weighting factors. If a threshold, 6, is imposed on the weighting factor, a teaching scheme can be executed to modify the class concept (15). For instance, one can assign the following simple rule for adaptive learning based on an updated weighting factor,

$$S_{ij} = \begin{cases} 1, & \text{if } W_{ij} \geq \theta \\ 0, & \text{if } W_{ij} < \theta. \end{cases}$$

A component of a D, assigned by name to a PU class, is inserted into the cell as non-essential if it fails to match any essential or non-essential components of the original class definition. If the new non-essential component occurs frequently in subsequent D's, its corresponding W_{ij} in the PU cell increases. When $W_{ij} > 0$, its status in the class definition becomes essential. This happens when the class concept of a certain object is changing with time.

In the case of self-organization, the pattern name might not be submitted to PPS. The generation of a class concept would then have to proceed through the identification phase. If a D matches all essential parts of a certain class pattern, it is assigned to the class. Additional information in that D then modifies the class by inserting non-essential components. If a certain component does not match in any PU cell, another class concept is introduced by the insertion of a new PU cell derived from the D.

Like other syntax-directed analysis schemes, PPS can also handle a recursive definition of some simple pattern classes. In fact, PPS not only enables class to be defined in a recursive generative manner, but also enables a recursive class definition to be generated from an object that is composed of recursive components. The general strategy for collapsing similar or recursive components is as follows:

a) Any connected set of components with a terminator (a component which is not directly connected with a component of greater hierarchal level than its own) is called a sub-universe of D.

b) Subuniverses are specified by the number of distinct hierarchical levels they contain and the hierarchical level of the terminator.

c) During pattern class generation, subuniverses are generated component by component starting from a terminator (which is recognized in D as a referee having a lower level referent than itself).

d) By noting similarity in the linkage, feature and junction fields of D, topological similarity of subuniverses can be recognized.

e) Topologically similar subuniverses which contain terminators of the same hierarchical level can be collapsed into a subuniverse of an identical structural class, allowing the inclusion of alternative terminators if necessary.

f) A pair of topologically similar subuniverses connected at corresponding components and whose terminators have different hierarchical levels is called a recursive realization.

g) By the use of certain rewriting rules, the pair displaying a recursive realization is combined into a recursive subuniverse with appropriate labels indicating a recursion pointer.

h) The same process is repeated until D cannot be reduced further.

i) The resulting D is then transferred to a pattern cell which is called a recursion pattern class.

Figure 12 shows diagrammatically how a Markov type transformation is performed on a simple recursive pattern. The transformation continues until an irreducible recursive pattern class is obtained. Figure 13a shows diagrammatically a recursive realization at a certain stage of transformation; Figure 13b shows how the D is partitioned into such a realization and how subsequent transformation is performed. (1* 1*) and (2 1*) are called a recursive pair, i.e., (2* 1*) will point to a component which can be recursively represented by the pattern identical to that subsequent to (1 1).

Conversely, the reverse productions can expand the recursive pattern class into an arbitrary version of a class,as well as recognizing objects that belong to the class.

Like all generative grammars, these production rules have limitations in defining a specific class. The continuous refinement of this concept is one of the key objectives of PPS.

The Function RECOGNIZE

When a scene is transformed into a D with both the FC and the JC derived for each component, a recalling process precedes the recognition and interpretation phases. The differential recalling process proceeds in the following manner:

a) Primitive forms (e.g., closed lines, open-ended lines) are first recorded for subsequent processing. Each primitive form can be a center of focal attention and a potential candidate for a nucleation component about which a recognizable subpattern can be generated.

b) The primitive form of the least hierarchical level is processed first and continually expands to neighboring components as the identification process goes on.

c) The FC of the reference component is used to extract a subset of PU by searching in parallel over the feature codes of the reference components of pattern cells. This yields a subset of PU for further matching.

d) Essential components, starting with those of least level, are then compared with the essential components of the pattern cells with respect to linkage, feature and junction fields until a smaller subset of pattern cells is determined. This process continues until all essential components of a single PU cell or a subset of PU cells match the corresponding components in the subpattern of the D. Thus, at this stage, pattern classes with identified topological similarity to some of the embedded objects in the D are identified. Further detailed search for local form match then continues.

e) The string code of each essential component is reconstructed using a hash code function. The string matching is performed by a scheme modified from the matching of genetic codes, developed by one of the authors (16)(17). Hence, both perfect match and partial match are performed. The former indicates the components are identical, while the latter implies the identification is fuzzy.

f) The same process is repeated for other primi tive forms unless they have been included in the essential part of the previously identified object. Components that do not belong to the non-essential part of the PU cell are considered as external to the object.

In the "apple-3" case, the apple outline and the character "3" are identified from their FC as primitive forms. Since the apple outline is at the least level, it is processed first and is found to match the essential component in the apple PU cell, whereas, the stem matches only the non-essential component and hence, is considered as part of the apple. The "3" is then identified in a like manner. The inersecting arc and the connector, in this case, do not match any of the non-essential parts in the apple PU cell or the "3" PU cell and will be considered as external and connecting. Thus, both the identification of embedded objects as well as the interpretation of their connections are complete.

Conclusions

This paper has presented both the conceptual framework and a partial implementation of PPS. Although crude patterns such as the "apple-3" have been processed, some of the processing methods and concepts await further refinement. The manipulation and reproduction phases (see Fig. 6) are based on the triangular mesh scheme. The former phase involves the construction of a set of functions that can deform or transform components of an object D. The latter phase then reproduces the results in another field matrix.

Although PPS is oriented towards developing computer visual perception, it should be mentioned that the Information structures and processing methodology developed can be applied to other data structures which are multi-linked.

Acknowledgement

This research is based on the doctoral dissertation of Edward L. Morofsky who was supported by the National Defense Education Act and the National Science Foundation at Carnegie-Mellon University.

References

(1) Feder, J. (1969) Linguistic specification and analysis of classes of line patterns. N. Y. U. Lab. Electroscience Research, Tech. Report 403-2.

(2) Shaw, A. C. (1970) Parsing of graph-representable pictures. Journal of A. C. M., July, 17, No. 3, 453-81.

(3) Narasimhan, R. (1966) Syntax-directed interpretation of classes of pictures. Comm. A. C. M., *9,* 166-73.

(4) Wong, A. K. C. and E. L. forofsky (1971) Automatic description, recognition and classification of complex line patterns, (draft in preparation).

(5) Neisser, U. (1966) Cognitive Psychology. Appleton-Century-Crofts, New York.

(6) Djang, S. (1937) The role of past experience in the visual perception of masked forms. J. Exp. Psychol., 20, 29-59.

(7) Minsky, M. (1961) Steps toward artificial intelligence. Proc. Inst. Radio Engr., 49, 8-30.

(8) Hochberg, J. E. (1964) Perception. Prentice-Hall.

(9) Pritchard, R. M., W. Heron and D. 0. Hebb (1960) Visual perception approached by the method of stabilized images. Canad. J. Psych., XIV, 67-77.

(10) Evans, C. R. (1968) Fragmentation of patterns occurring with tachistoscopic presentation. IEEE Conf. Publ., 4£, 250-63, 1968.

(11) Buswell, G. T. (1935) How We Look at Pictures. Univ. Chicago Press.

(12) Hochberg, J. (1962) The psychophysics of pictorial perception. Audio-Vis. Comm. Rev., JL0, 22-54

(13) Attneave, F. (1954) Some informational aspects of visual perception. Psych. Rev., 61., 183-193.

(14) Gibson, E. J. (1963) Perceptual learning. Annu. Rev. Psych. , 14, 29-56.

(15) Nilsson, N. J. (1965) Learning Machines. McGraw-Hill.

(16) Reichert, T. A. and A. K. C. Wong (1970) An application of information theory to genetic mutations and the matching of polypeptide sequences. (Submitted to J. Theoret. Biol.).

(17) Wong, A. K. C., T. A. Reichert and B. 0. Aygun (1971) A generalized method for matching amino acid and unambiguous genetic code sequences. (Submitted to J. Computers in Med. and Biol.).

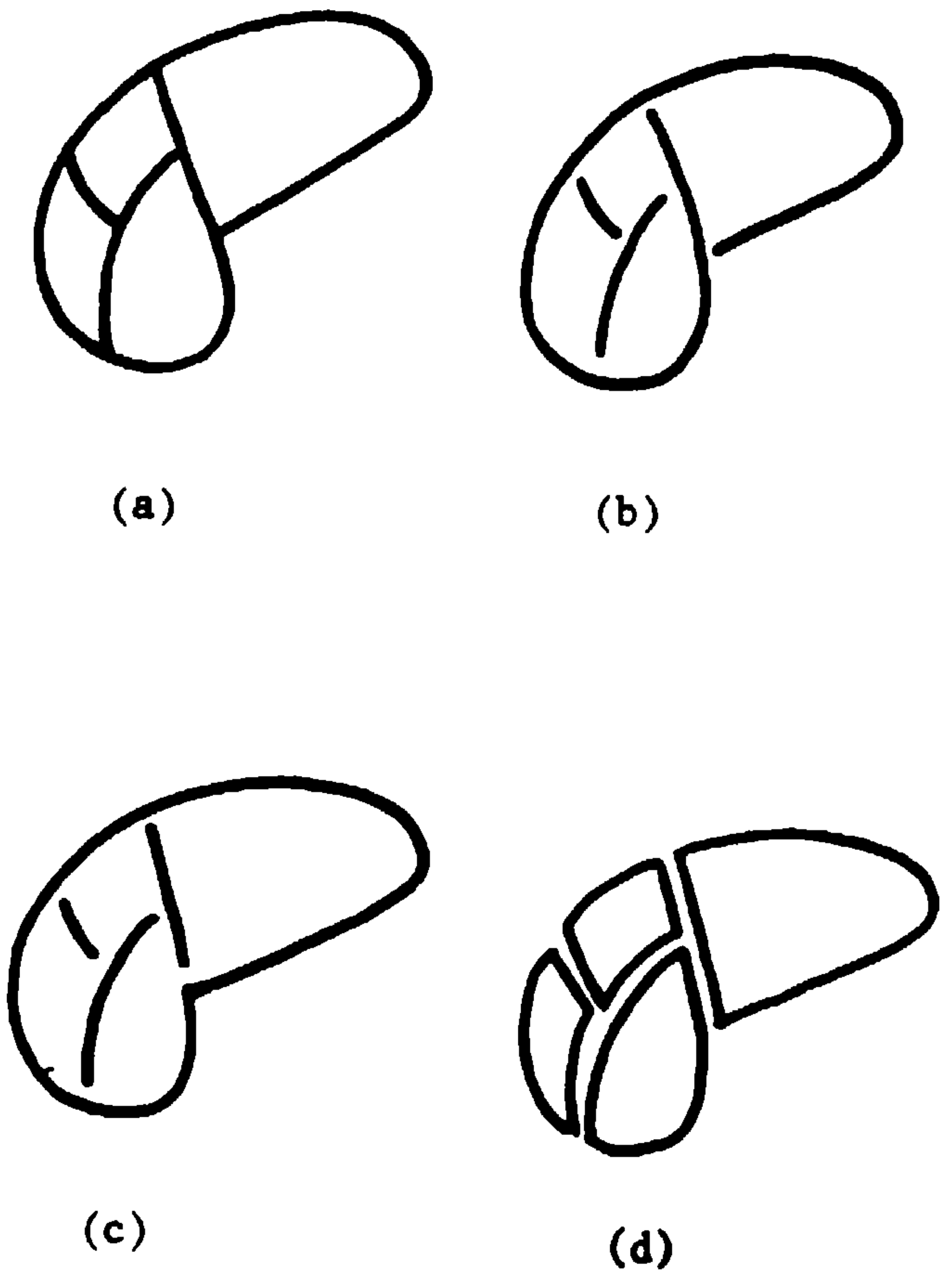


Figure 1.

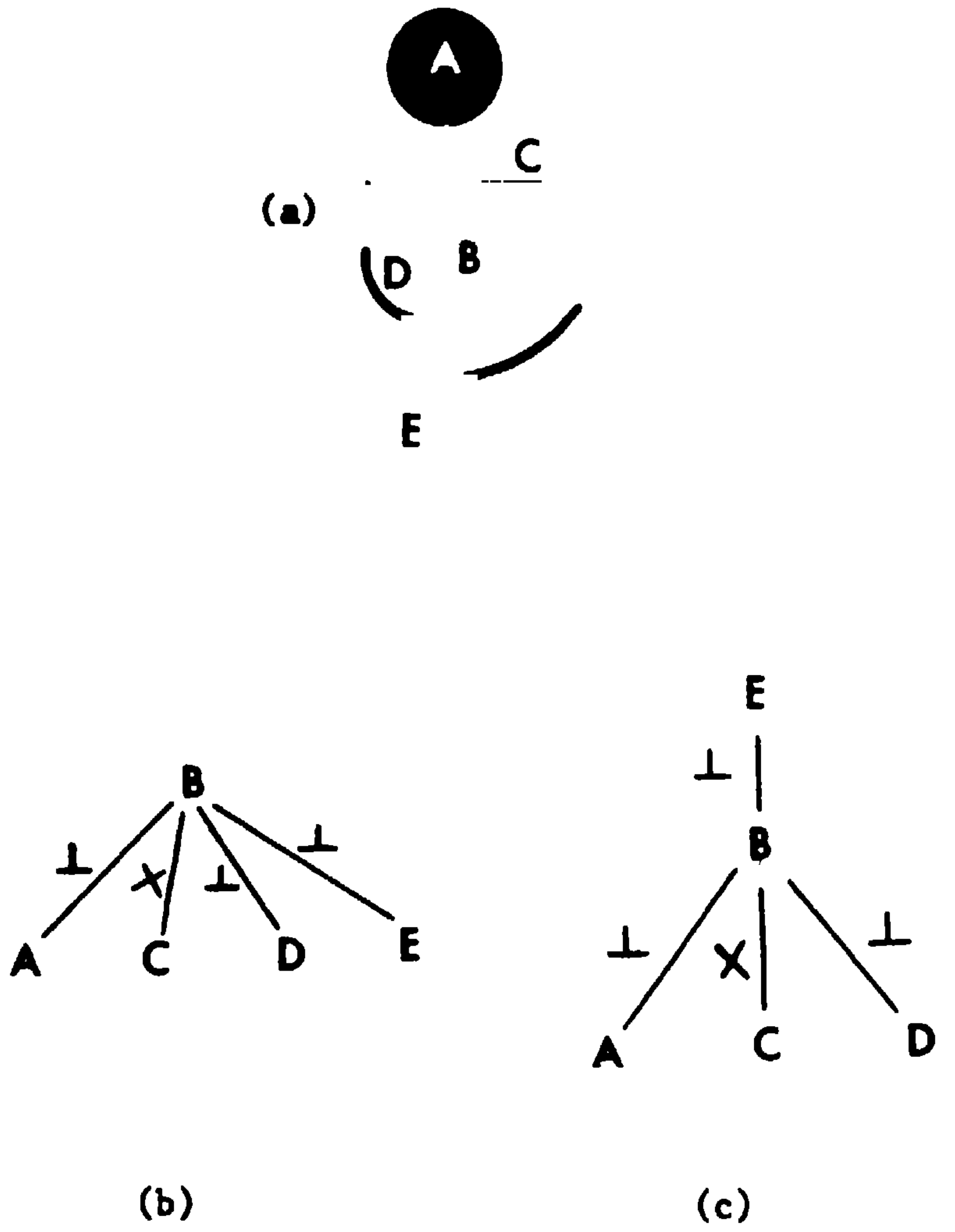


Figure 2

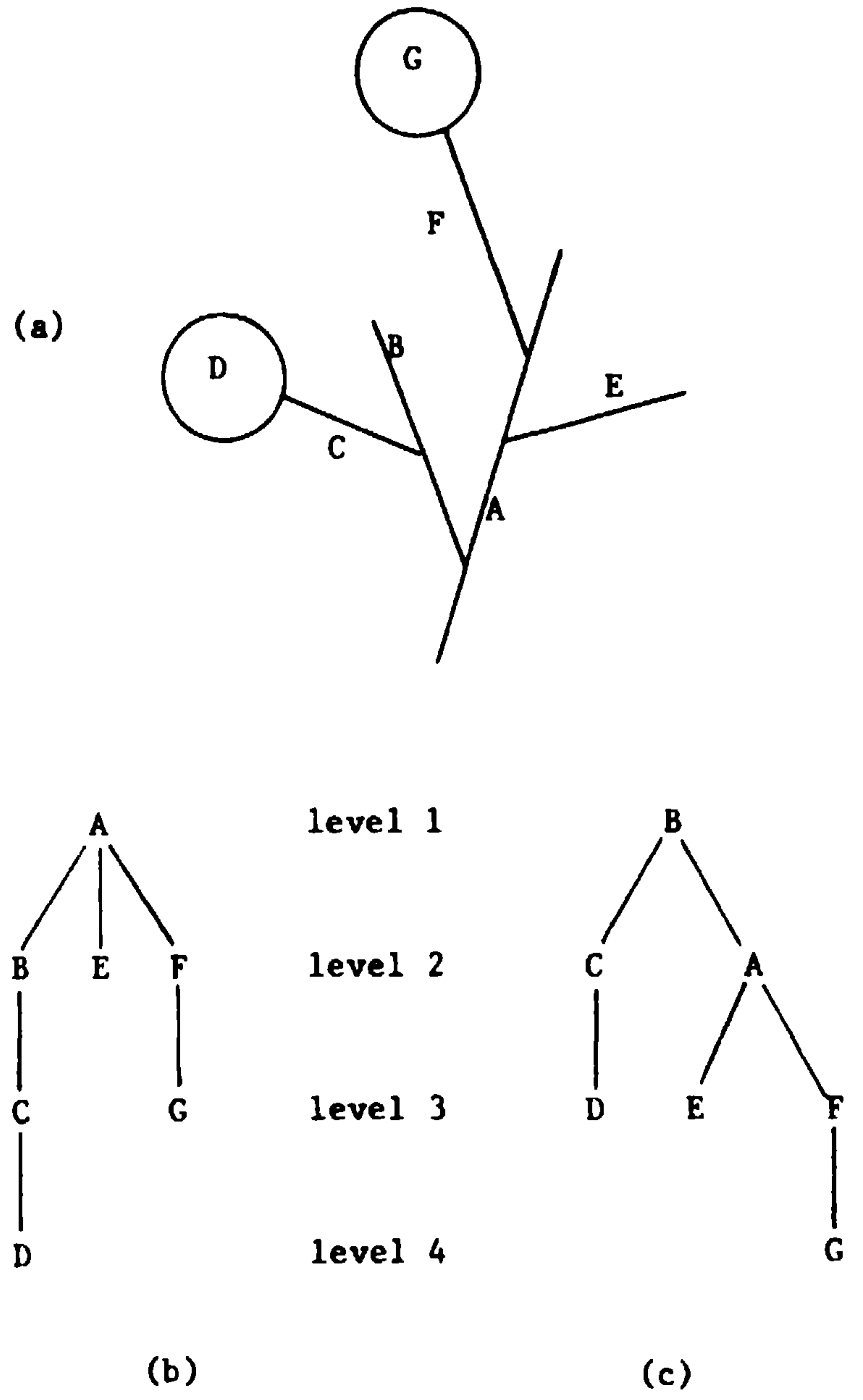


Figure 3.

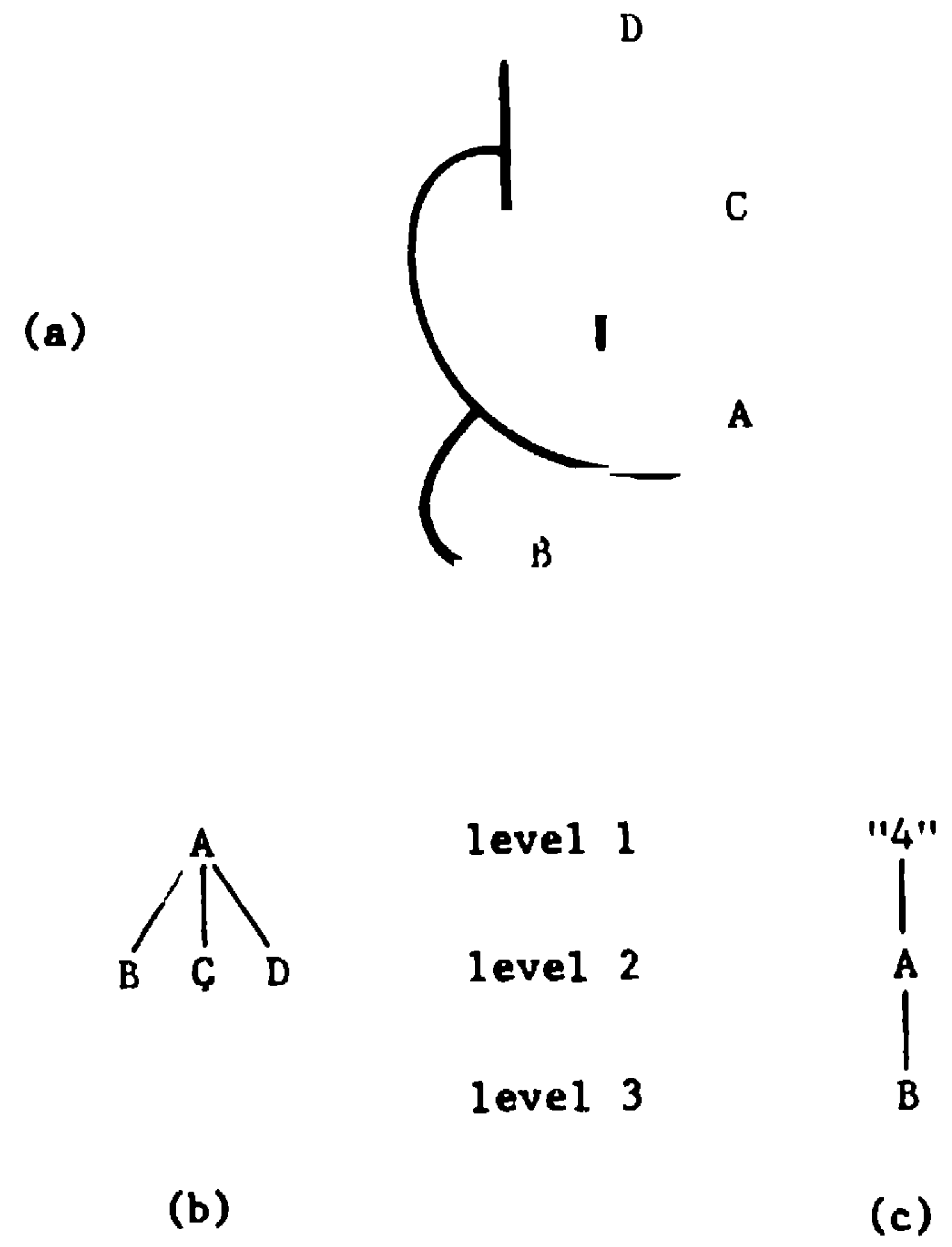


Figure 4.

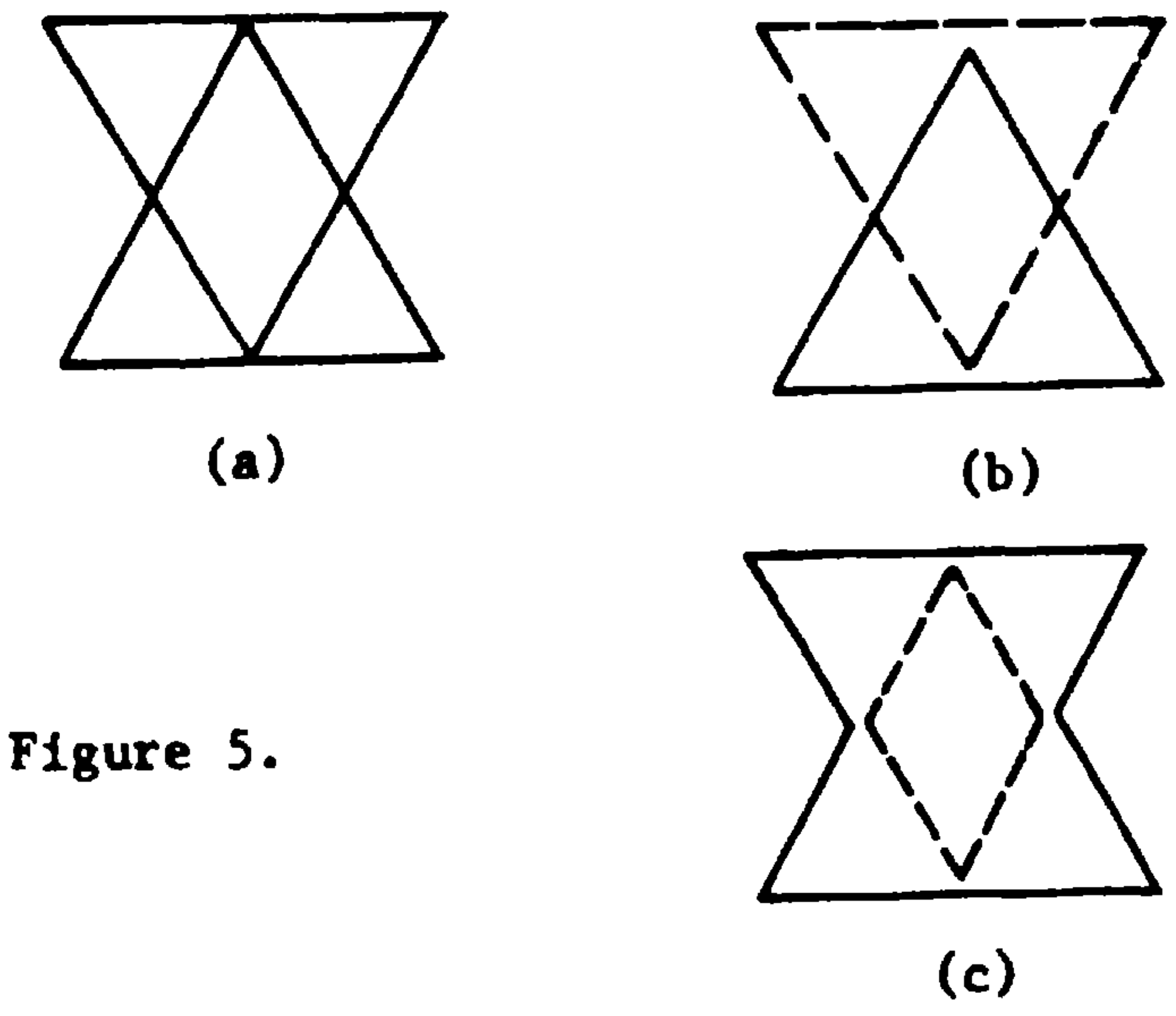


Figure 5.

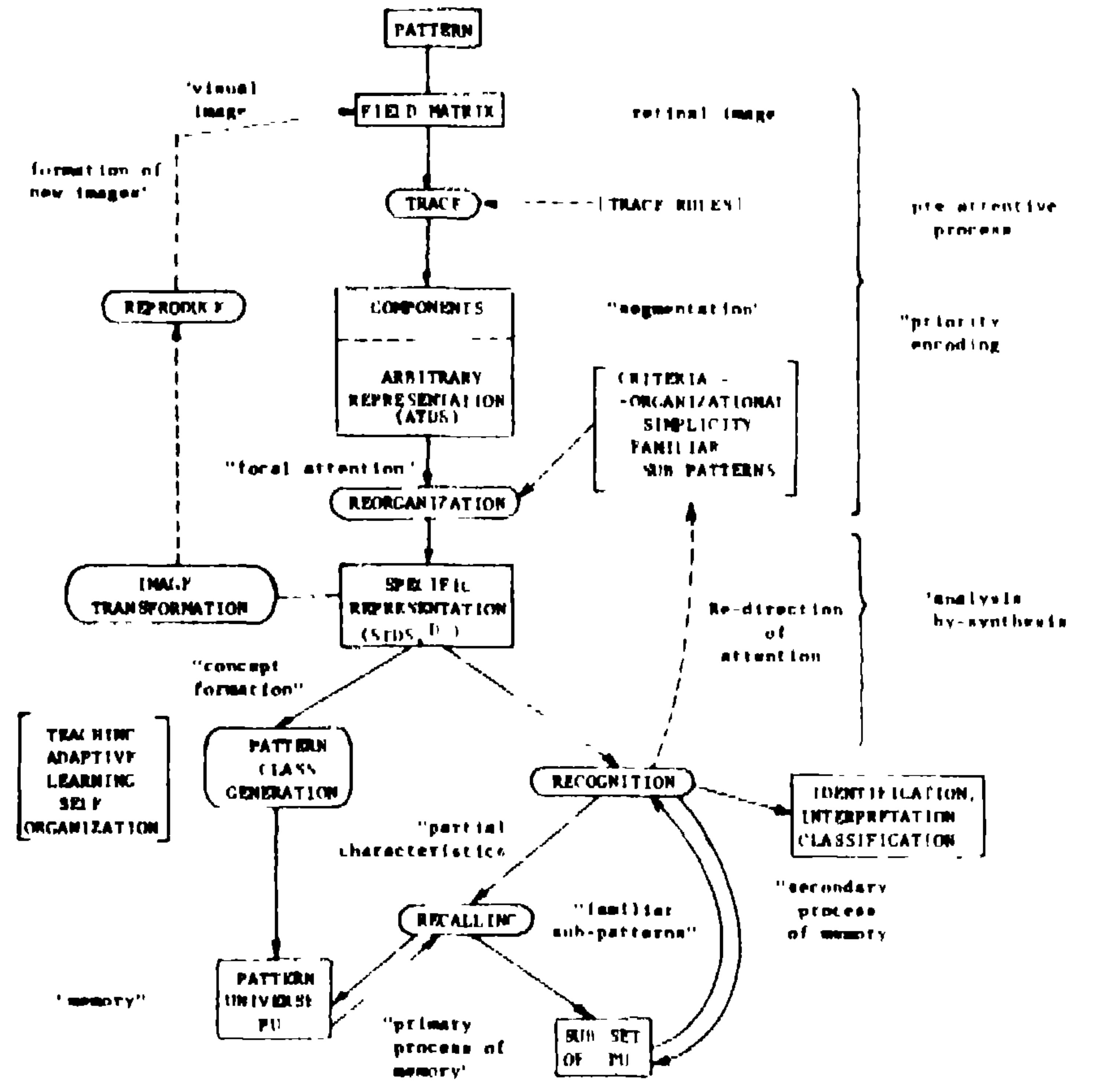
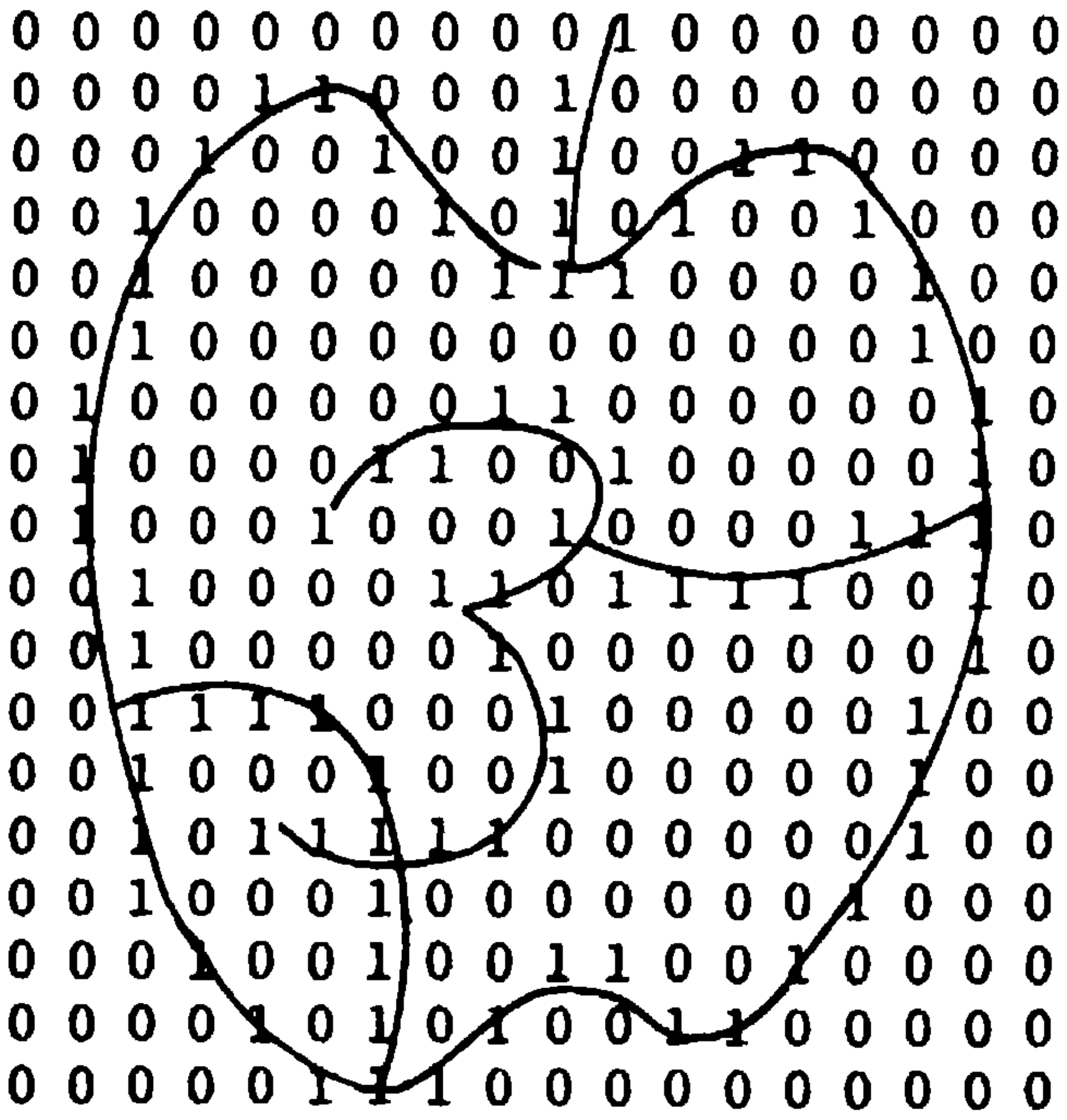


Figure 6.



```
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 0 0 1 0 0 1 1 0 0 0 0
0 0 1 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0
0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0
0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0
0 0 1 0 0 0 0 1 1 0 1 1 1 1 0 0 1 0
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0
0 0 1 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0
0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0
0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 0
0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
0 0 0 1 0 0 1 0 0 1 1 0 0 1 0 0 0 0
0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 0 0 0
0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
```

Figure 7

```
0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0
0 0 0 0 3 3 0 0 0 2 0 0 0 0 0 0 0 0
0 0 0 3 0 0 3 0 0 2 0 0 3 3 0 0 0 0
0 0 3 0 0 0 0 3 0 2 0 3 0 0 3 0 0 0
0 0 3 0 0 0 0 0 3¯2 3 0 0 0 0 3 0 0
0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0
0 3 0 0 0 0 0 0 4 4 0 0 0 0 0 0 3 0
0 3 0 0 0 0 4 4 0 0 4 0 0 0 0 0 3 0
0 3 0 0 0 4 0 0 0¯6 0 0 0 0 5 5¯3 0
0 0 3 0 0 0 0 4 4 0 5 5 5 5 0 0 3 0
0 0 3 0 0 0 0 0 4 0 0 0 0 0 0 0 3 0
0 0¯5 6 6 6 0 0 0 4 0 0 0 0 0 3 0 0
0 0 3 0 0 0 6 0 0 4 0 0 0 0 0 3 0 0
0 0 3 0 4 4¯7 4 4 0 0 0 0 0 0 3 0 0
0 0 3 0 0 0 6 0 0 0 0 0 0 0 3 0 0 0
0 0 0 3 0 0 6 0 0 3 3 0 0 3 0 0 0 0
0 0 0 0 3 0 6 0 3 0 0 3 3 0 0 0 0 0
0 0 0 0 0 3¯4 3 0 0 0 0 0 0 0 0 0 0
```

Figure 8

11: 3 4 4 4 3 -2 3 2 2 3 4 4 5 4 5 5 -3 5 5 6 5 5 6 6 6 7 8 7 6 6 7 -4 7 8 8 8 1 1 1 -5 1 1 8 1 1 2 (apple outline)

21: -4 1 1 1 1 -7 1 8 7 7 7 -5 (intersecting arc)

22: -2 1 1 1 2 (stem)

23: -3 7 7 6 7 7 7 8 -6 (connector)

31: 3 3 -7 3 3 2 1 8 8 3 2 -6 2 8 7 6 7 6 ("3")

Figure 9.

linkage field				feature code	junction code	hash code	
referee		referent					
1	1	2	1	70	66	1286	(apple outline)
1	1	2	1	70	66	1286	
1	1	2	2	70	14	1286	
1	1	2	3	70	22	1286	
2	1	3	1	5	5	241	(arc)
2	2	1	1	5	2	723	(stem)
2	3	3	1	7	2	791	(connector)
3	1	0	0	55	0	801	("3")

Feature code:

Closed	- 2
Straight segment	- 3
Positive or convex curve	- 5
Negative or concave curve	- 7
Angle	- 11
Intersecting	- 13

Junction code:

T-junction	- 2
Double	- 3
Intersection	- 5
Outside of or positive side of	- 7
Inside of or negative side of	- 11
K-junction	- 13

Figure 10. The description (D) for the "apple-3" drawing with feature code and junction code definitions.

	class name in hash code	referee label		referent label		feature code	junction code	component status	weighting factor	hash code
Apple	777	1	1	2	1	70	14	1	1	1286 - outline
		2	1	1	1	5	2	0	.2	723 - stem
"3"	211	1	1	0	0	55	0	1	1	801
Tree	621	1^{*+}	1^{*+}	2^{*+}	1^{*+}	3	2	1	1	1623
		1^{+}	1^{+}	2^{+}	2^{+}	3	2	1	1	777
		2^{+}	2^{+}	1^{+}	1^{+}	3	2	0	.6	207

* - indicates recursive component pair.
\+ - indicates additional components are possible.

Figure 11. Pattern Universe (PU) containing pattern class definitions of apple, "3" and tree.

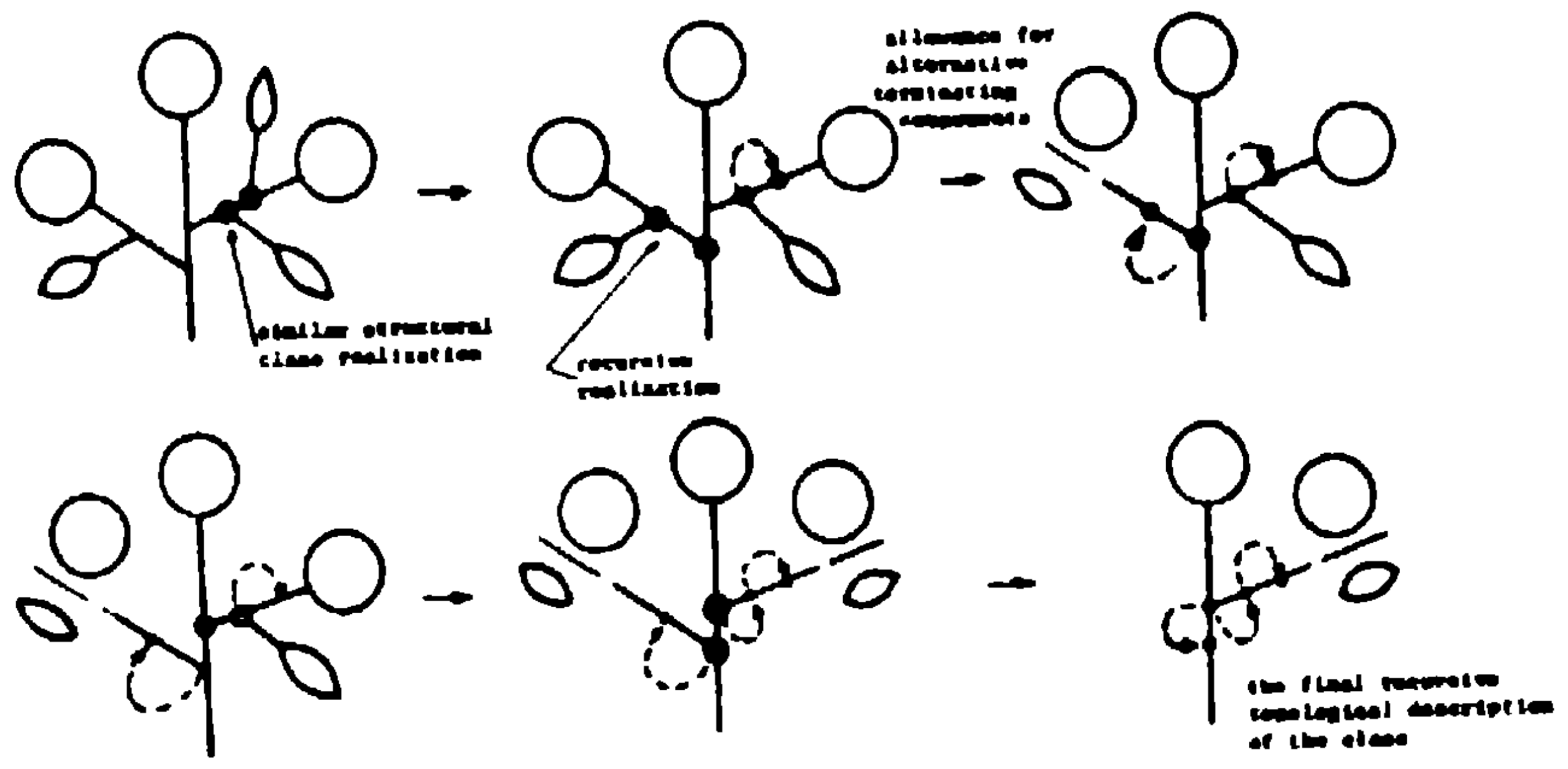
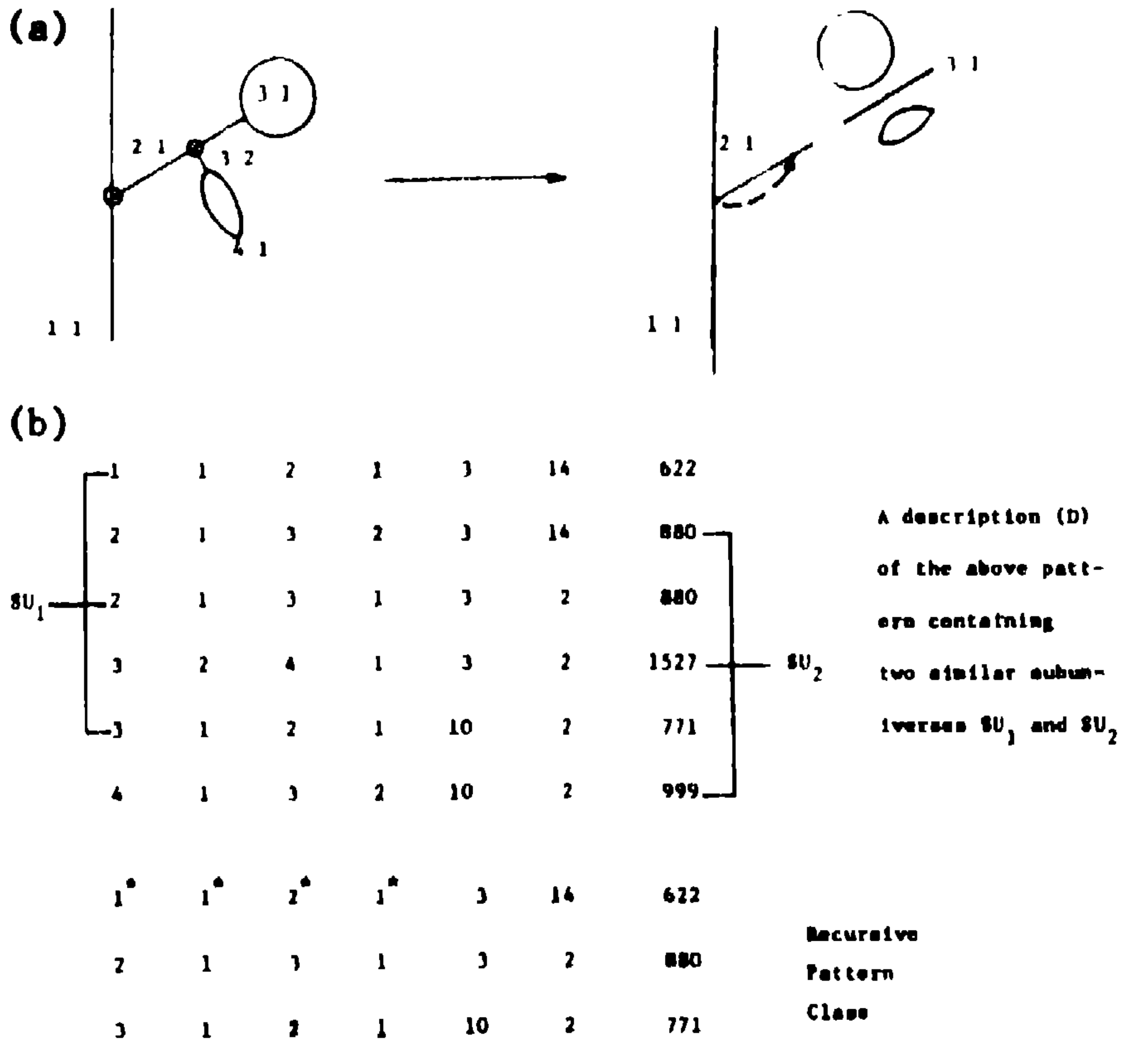


Figure 12.



1	1	2	1	3	14	622
2	1	3	2	3	14	880
2	1	3	1	3	2	880
3	2	4	1	3	2	1527
3	1	2	1	10	2	771
4	1	3	2	10	2	999

A description (D) of the above pattern containing two similar subuniverses SU_1 and SU_2

1^{*}	1^{*}	2^{*}	1^{*}	3	14	622
2	1	3	1	3	2	880
3	1	2	1	10	2	771

Recursive Pattern Class

Figure 13. The description (D) of the drawing of Fig. 13a is given in Fig. 13b with the two similar subuniverses SU_1 and SU_2 forming a recursive realization. D is collapsed into a recursive pattern class (Fig. 13b).