KNOWLEDGE STRUCTURES FOR COMMUNICATIONS IN HUMAN-COMPUTER SYSTEMS

GENERAL AUTOMATA-BASED

Eldo C. Koenig

IEEE computer society
60th anniversary

WILEY-INTERSCIENCE
A John Wiley & Sons, Inc., Publication
KNOWLEDGE STRUCTURES FOR COMMUNICATIONS IN HUMAN-COMPUTER SYSTEMS
Press Operating Committee

Chair
Roger U. Fujii,
Vice President
Northrop Grumman Mission Systems

Editor-in-Chief
Donald F. Shafer
Chief Technology Officer
Athens Group, Inc.

Board Members
Mark J. Christensen, Independent Consultant
Herb Krasner, President, Krasner Consulting
Ted Lewis, Professor Computer Science, Naval Postgraduate School
Hal Berghel, Professor and Director School of Computer Science, University of Nevada
Phillip Laplante, Associate Professor Software Engineering, Penn State University
Richard Thayer, Professor Emeritus, California State University, Sacramento
Linda Shafer, Professor Emeritus, University of Texas at Austin
James Conrad, Associate Professor, UNC-Charlotte
Deborah Plummer, Manager-Authored books

IEEE Computer Society Executive Staff
David Hennage, Executive Director
Angela Burgess, Publisher

IEEE Computer Society Publications
The world-renowned IEEE Computer Society publishes, promotes, and distributes a wide variety of authoritative computer science and engineering texts. These books are available from most retail outlets. Visit the CS Store at http://computer.org/cspress for a list of products.

IEEE Computer Society / Wiley Partnership
The IEEE Computer Society and Wiley partnership allows the CS Press authored book program to produce a number of exciting new titles in areas of computer science, computing and networking with a special focus on software engineering. IEEE Computer Society members continue to receive a 15% discount on these titles when purchased through Wiley or at wiley.com/ieeecs

To submit questions about the program or send proposals please e-mail dplummer@computer.org or write to Books, IEEE Computer Society, 100662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314. Telephone +1-714-821-8380.

Additional information regarding the Computer Society authored book program can also be accessed from our web site at http://computer.org/cspress
KNOWLEDGE STRUCTURES FOR COMMUNICATIONS IN HUMAN-COMPUTER SYSTEMS

GENERAL AUTOMATA-BASED

Eldo C. Koenig
To Gloria
## CONTENTS

**Preface** xi  

1. **Introduction** 1  
   1.1 Considerations for Establishing Knowledge Structures for Computers 1  
   1.2 Knowledge About Automata as a Subset of World Knowledge 2  
      1.2.1 General Automata 4  
      1.2.2 Extracting and Storing the Meanings of Sentences 6  
      1.2.3 Associating Knowledge 9  
      1.2.4 Establishing Conclusions and Inferences 14  
   Exercises 17  

2. **A General Automaton** 19  
   2.1 Formal Analysis for a General Automaton 19  
      2.1.1 General Analysis 19  
      2.1.2 Graph Model 28  
      2.1.3 Select Properties of the Graph Model 41  
   2.2 An Application of the Disciplines to the Modeling of Natural Automata 48  
      2.2.1 A Case Study 49  
      2.2.2 Required State Changes 52  
      2.2.3 Algorithm for Determining Required State Changes 58  
   Exercises 60  

3. **A General Automaton: Detailed Analysis** 63  
   3.1 Distinguishable Receptors and Effectors 63  
   3.2 Nonhomogeneous Environments 67  
   3.3 Transformation Response Components 70  
   3.4 Nonshared Environments Interpreted as Distinguishable 71  
      3.4.1 Model for Performance in Both Shared and Nonshared Environments 72  
      3.4.2 Model for Performance in Shared Environments 77  
   Exercises 79
4. Processing of Knowledge About Automata 81

4.1 Formulation of a Language Information Theory 82
  4.1.1 Class 1 Sentence 86
  4.1.2 Class 2 Sentence 89
  4.1.3 Class 3 Sentence 93
  4.1.4 Class 4 Sentence 94
  4.1.5 Class 5 Sentence 98
  4.1.6 Class 6 Sentence 101
  4.1.7 Class 7 Sentence 102

4.2 Extracting and Storing the Meaning of Sentences by Computer 103
  4.2.1 Description of an Algorithm 104

4.3 Knowledge Association 105
  4.3.1 Association by Combining Graphs Through Common Points 108
  4.3.2 Associations by Combining Graph \((n+1)\)-Tuples 110
  4.3.3 Computer Methods for Association of Knowledge 112

4.4 Deductive Processes 113
  4.4.1 Deductive Processes Related to Association Through Common Points 113
  4.4.2 Deductive Processes Related to Association by Combining Graph Tuples 117
  4.4.3 Deductive Processes with Aristotelian Form A as a Premise 120

4.5 Inferences 124
  4.5.1 Inferences Related to a Single Graph Tuple of Associated Knowledge 124
  4.5.2 Inferences Related to More than One Graph Tuple of Associated Knowledge 134

Exercises 136

5. A General System of Interactive Automata 139

5.1 Formal Analysis for a General System of Interactive Automata 139
  5.1.1 General Analysis 139
  5.1.2 Microsystem Model 149
  5.1.3 Macrosystem Model 156

5.2 Example Applications 162
  5.2.1 A Two-Component System 162
  5.2.2 A System of Many Components 171

Exercises 173
<table>
<thead>
<tr>
<th>6. Processing of Knowledge About Systems of Automata</th>
<th>175</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 A General System of Interactive Automata: Detailed Analysis</td>
<td>175</td>
</tr>
<tr>
<td>6.1.1 The Microsystem Model</td>
<td>176</td>
</tr>
<tr>
<td>6.1.2 The Macrosystem Model</td>
<td>176</td>
</tr>
<tr>
<td>6.2 Knowledge Structures for Sentences Describing Systems of Interactive Automata</td>
<td>177</td>
</tr>
<tr>
<td>Exercises</td>
<td>184</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. Changing Expressions of Knowledge for Communication from One Form and Style to Another</th>
<th>185</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Introduction</td>
<td>185</td>
</tr>
<tr>
<td>7.2 Sets and Relations</td>
<td>187</td>
</tr>
<tr>
<td>7.3 Establishing Open Expressions and Open Sentences</td>
<td>189</td>
</tr>
<tr>
<td>7.4 Selecting Subsets of Open Expressions</td>
<td>193</td>
</tr>
<tr>
<td>7.5 Applying the Results of the Above Analysis</td>
<td>195</td>
</tr>
<tr>
<td>7.6 Summary and Conclusions</td>
<td>199</td>
</tr>
<tr>
<td>Exercises</td>
<td>202</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. Electronic Security Through Pseudo Languages</th>
<th>203</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Introduction</td>
<td>203</td>
</tr>
<tr>
<td>8.2 Definitions, Sets, and Relations</td>
<td>204</td>
</tr>
<tr>
<td>8.3 Analysis for E-Security Through Pseudo Languages</td>
<td>205</td>
</tr>
<tr>
<td>8.3.1 A Basic E-Security System</td>
<td>205</td>
</tr>
<tr>
<td>8.3.2 A Two-Step Encryption System</td>
<td>208</td>
</tr>
<tr>
<td>8.3.3 E-Signing</td>
<td>212</td>
</tr>
<tr>
<td>8.4 Summary and Conclusions</td>
<td>213</td>
</tr>
<tr>
<td>Exercises</td>
<td>213</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix A: Analysis for an Effective Operation of a General Automaton</th>
<th>215</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 Introduction</td>
<td>215</td>
</tr>
<tr>
<td>A.2 Recursive Methods</td>
<td>216</td>
</tr>
<tr>
<td>A.3 Effective Operation Analysis</td>
<td>236</td>
</tr>
<tr>
<td>Exercises</td>
<td>241</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix B: Analysis for an Effective Operation of a General System of Interactive Automata</th>
<th>243</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1 Introduction</td>
<td>243</td>
</tr>
<tr>
<td>B.2 Microsystem Graphs</td>
<td>248</td>
</tr>
</tbody>
</table>
The book presents knowledge structures for communications in human-computer systems (HCS) based on general automata. The general automaton was considered basic in disciplining the natural language, in that knowledge to be communicated is about automata and histories of automata. The work of others on finite automata does not include the detail required for finite general automata. To accomplish completeness and unification for the broad concept of a general automaton, the analysis applies the algebra of sets and functions and follows with the application of combinatorial mathematics for graph theory to define a general graph model for knowledge representation. Since interacting automata must be interfaced in space and time, the graph model is fundamental to the analysis of systems of automata and is supported by more than 50 theorems and proofs. The material presented makes reference to 34 publications by the author or jointly by the author and graduate students.

The resulting model provides knowledge representations for software engineering. Of the many features required of a method to achieve the desired communication in HCS, six are identified and illustrations presented for achieving them by the general automata method (GAM):

1. Extracting and storing the knowledge of sentences
2. Knowledge association
3. Deductive processes
4. Inferences
5. Feedback
6. Sequencing of knowledge

After the analysis for each feature is presented, the result is illustrated with practical applications. Algorithms and programs are described in achieving some of the features, and additional algorithms and further research are indicated.

The material is presented in the style and form that is suitable for either an academic or an industrial setting, and for either self-study or group learning. It has been taught to advanced undergraduate and graduate students. For those with interest and background in applied computer science and software engineering, the book describes algorithms and programs, and suggests others.
Emphasis for this group would be placed on Chapters 1, 4, 6, 7, and 8. Some knowledge of algebraic language and systems programming, algebra of sets and functions, and combinatorial mathematics for graph theory would be helpful. For those with interest in research and advanced development, the book supplies guiding principles and suggests additional research, as in e-translation and human-computer interactions. Emphasis for this group would be placed on Chapters 2, 3, 5, and the Appendices. Some knowledge of algebra of sets and functions, combinatorial mathematics for graph theory, recursive theory, and logical foundations is desirable.

The author acknowledges the various support efforts of a number of groups at the University of Wisconsin–Madison, especially the Computer Sciences Department, the Electrical and Computer Engineering Department, and the School of Education in making available the time and the facilities for establishing many of the results presented. The author also acknowledges and thanks the many groups and individuals throughout the world for their expressed interest and approval of the work as it progressed over the years. Also acknowledged and appreciated are the efforts of the author’s wife, Gloria, in putting the material of this book in its final form and style for publication.

Department of Computer Sciences  
University of Wisconsin–Madison  
(Retired status)  

Eldo C. Koenig, Ph.D.
CHAPTER 1

Introduction

1.1 CONSIDERATIONS FOR ESTABLISHING KNOWLEDGE STRUCTURES FOR COMPUTERS

Thousands of languages have been developed by as many isolated societies over thousands of years (Wuethrich 2000). The object of any society in developing a language is to communicate perceived knowledge about its environment. Environments are considered to contain living things (automata) that perform individually and in groups (interactive automata) and that make changes in the environments through their responses (histories of automata). That is, perceived knowledge is about automata and histories of automata, and languages are established to communicate the knowledge.

A model for communication in human-computer systems involves more than a syntactical analysis for extracting the meanings of sentences as is demonstrated by the following two pairs of sentences:

“John struck the table with a hammer.”
“John struck the table with a glass top.”
“The deer came out of the wood.”
“The worm came out of the wood.”

A model should also accommodate the duplication of the following example performances:

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

   “Mrs. Bee and Mrs. Cee shop at Supermarket Dee.”

2. An observer one day reads that during the Renaissance period, the people of Venice built the Library of St. Mark, and years later reads that
during the Renaissance period, the Romans built the Church of St. Peter, and then says,

“During the Renaissance Period, the people of Venice built the Library of St. Mark, and the Romans built the Church of St. Peter.”

3. A person at one time hears the sentence

“John ate a nut.”

and at a later time says to a person of little experience,

“John ate a nut. He put it into his mouth. He chewed the nut with his teeth and swallowed it.”

4. An observer sees and almost simultaneously hears the oral sound of an animal one day, hears and almost simultaneously smells (but does not see) that same animal at some later time, and still later, only smells that animal, and then draws a picture of that animal in answer to the question

“What is that?”

5. A person hears the statement

“Copper is a metal.”

and then responds with the statement

“Copper is malleable.”

without having previously received that responding statement.

6. A person hears the statement

“Mary put the nut into her mouth.”

and then responds with the statement

“Mary probably chewed the nut.”

1.2 KNOWLEDGE ABOUT AUTOMATA AS A SUBSET OF WORLD KNOWLEDGE

The above discussion presents some important considerations for establishing a model for knowledge structures for computers in application to highly interactive human-computer systems. The desire is to replace a human with a
computer in an interactive system and have it perform in the same manner. To achieve a high degree of success, one must discover the order and organization of the universe that a human wishes to describe in the natural language. A force opposed to order is the desire for quick use of a computer.

Any degree of order suggests a mathematical model for knowledge structures for computers. One advantage of a mathematical model involving functions is that knowledge to be stored may be restricted to that which meets the requirements for single-valueness of the functions. To a casual observer, it is reasonable to expect that a computer operating on these principles may appear superior in intelligence compared with a human who is not so rigidly disciplined.

A mathematical model for knowledge structures in the form of a graph is very desirable. One can make use of many existing graph properties by properly interpreting them in physical terms. There are also many existing algorithms associated with graphs for calculating properties.

The technological advances in hardware are encouraging highly interactive human-computer systems that involve a large amount of memory and computing. Parsing methods may be giving way to template matching methods or to a combination of the two. Parsing methods favor single processors while template matching methods favor parallel processors. It is economically practical to have many microprocessors operating in parallel (Koenig 1994b). A million processors in an array is not considered outrageous (Cray 1978).

The subset of world knowledge that has been chosen for consideration in establishing a model for knowledge structures for computers is knowledge about automata. It represents order and organization of the universe. To the author’s knowledge, no other model for knowledge structures uses the order of automata to describe knowledge, and no other model employs the mathematical treatment presented here.

Briefly, an automaton is defined as anything that can move or act of itself. For the past several centuries, society has witnessed various individuals and groups in their efforts to study and produce automata (Chapuis and Droz 1958, Sabliere 1966, Von Neumann 1961). In the 17th century, lower animals were viewed as natural automata by Descartes, and mechanical devices reflecting the technology of the times were viewed as artificial automata. Leibnitz regarded clockwork as the model for all automata, and, as a result, automata was not considered to be affected by or to affect the outside world. Beginning the 19th century, both natural and artificial automata were studied in relation to energy. An automaton at that time was considered a “heat engine burning some combustible fuel instead of the glycogen of the human muscles” (Wiener 1948). Babbage, early in the 19th century, had the first idea of a computing automaton.

The 20th century has witnessed a broad concept of automata viewed in total in its interaction with its environment. The concept played an important role in the natural sciences in a qualitative sense at the beginning of the century. In the 1920s, Bush (1929) established the first computing automata of the analog class operating on the continuous principle, and, in the 1930s, Turing
(1936) did pioneering work on the digital class built around the all-or-none concept and operating at discrete moments of time. Prior to the decade of the 1960s, the concern was generally for computing automata as it performed independent of other automata. Since then, interest grew rapidly in interacting automata operating in real time. Systems of non-computing artificial automata operating in the manufacturing processes, called automation, have been made to interact with systems of computing automata. Also prevalent today are interactive computing automata and interactive human-computer systems. It is in highly interactive human-computer systems where models for knowledge structures for computers become particularly important. Here, knowledge about automata will encompass knowledge about general automata, general systems of interactive automata, and about histories of general automata (Koenig 1972a, 1997b).

1.2.1 General Automata

Previous analysis for finite automata did not include the detail required for finite general automata. Normally, two equations are used to describe finite automata, and the state-transition diagram is used to study the operations. Twelve equations are required to describe finite general automata, general systems of interactive automata, and histories of general automata. This set of equations is established and discussed in Chapters 2, 3, and 5. The analysis takes into account component responses, distinguishable receptors and effectors, nonhomogeneous environments, and responses interpreted as stimuli for the five senses. Three graphs are used to study the operations of a general automaton and a general system of interactive automata. They are

Figure 1.1 General automaton operation: (a) diagram of operation of a general automaton; (b) environment graph containing the information of (a) and the corresponding graph tuple.
1. Processor graph with points representing states (extended state-transition diagram)
2. Environment graph with points representing environments (stimulus locations)
3. Time graph with points representing times

The operation of a general automaton is diagrammed to operate as shown in Figure 1.1a and is described as follows: Given a discrete moment $m+i$, at time $t_i$, a general automaton $a$ is accessing one and only one environment $e_i$ (stimulus location), receiving one and only one stimulus $s_i$ through one and only one auxiliary receptor $x_{i}''$ and only one principal receptor $x_{i}'$, and is in one and only one state $q_i$. Multiple accesses of a same environment during a period of consideration can be made only by the subject automaton. (For a system of automata, multiple accesses of a same environment can be made by different component automata of the system.) Furthermore, during the next moment $m+(i+1)$, at the time $t_{i+1}$, the automaton $a$ produces one and only one response $r_{i+1}$ through one and only one principal effector $z_{i}'$ and one and only one auxiliary effector $z_{i}''$ based on the aforementioned conditions. The response $r_{i+1}$ consists of a transformation response $f_{i+1}$, a spatial change response $d_{i+1}$, and a time change response $c_{i+1}$. The transformation response consists of responses interpreted as stimuli for sight $f_{si+1}$, smell $f_{mi+1}$, hearing $f_{hi+1}$, touch $f_{ti+1}$, taste $f_{ai+1}$, and is recorded as a response $b_{i+1}$ at the environment $e_i$ accessed at the first moment $m+i$. At moment $m+(i+1)$, the automaton is in state $q_{i+1}$ and at environment $e_{i+1}$.

If the automaton is removed from the diagram of Figure 1.1a and if the information it contains is included in the label for the connecting line of two points $e_i$ and $e_{i+1}$, Figure 1.1b results. This is the basic automaton graph of two points and an adjacent line, which is the environment graph for storing knowledge. The corresponding graph tuple is also shown in Figure 1.1b. The time graph is the same except the points represent times, and the times $t_i$ and $t_{i+1}$ of the line label are replaced with the environments $e_i$ and $e_{i+1}$. For the processor graph, the points represent states, and the number of terms of the line label is reduced.

The sequence of terms of the line label of each type of graph is not arbitrary but is determined by a graph function that is discussed in Chapters 2, 3, and 5. The requirement for single-valueness of the function must always be met. There are features like this resulting from the rigorous treatment of the problem of knowledge structures based on general automata that should give a computer following these principles much greater power than a human for displaying intelligence.

Feedback can be readily represented by the graph model for knowledge structures. Since the spatial change response component $d_{i+1}$ can be zero, an automaton can be told of its immediate past performance; i.e., it can interpret its own response as a stimulus from the current environment before receiving
a stimulus from a new environment. This immediate feedback is represented in an environment graph by loops. A detailed analysis of the graph model for feedback is presented in Section 3.4.

This model for knowledge structures for computers, based on general automata, accommodates procedures for

1. extracting and storing the meanings of sentences
2. associating knowledge
3. establishing conclusions and inferences

These are discussed in the following sections. It will also be shown how the model for knowledge structures accommodates the duplication of the example performances listed in Section 1.1.

1.2.2 Extracting and Storing the Meanings of Sentences

The meanings of sentences can be stored in the environment, time, and processor graphs. A graph of two points and an adjacent directed line is normally required to store the meaning of a sentence. An environment graph of two points and an adjacent directed line and a corresponding graph tuple are shown in Figure 1.1b. Not all of the automaton terms of a graph tuple will have word values when the meaning of a single sentence is stored. Each of the following sentences is an example of a class of sentence whose meaning can be extracted and stored as a knowledge structure based on general automata. The seven classes of sentences are discussed in Section 4.1.

1. “John struck the table.”
2. “Today, John drove the car carefully three blocks from the house to the store in five minutes.”
3. “John became angry.”
4. “The table has a glass top.”
5. “Today, Jim saw the dent in the table that John made yesterday.”
6. “Today, Jim read that John struck the table with a hammer yesterday.”
7. “Today, Jim wrote that John struck the table with a hammer yesterday.”

Consider, for example, the knowledge structure for the first sentence:

1. “John ($a$) struck ($f_{i+1}$) the table ($e_i$)”

Behind each content word (knowledge element) there is written an automaton term in parentheses that will take on that word as a value in the knowledge
structure. The meaning of the sentence is stored in the 16-tuple for the environment graph of Figure 1.1b in the following manner:

\[(e_{i}=\text{table}, - , - , - , - , - , a=\text{John}, - , f_{i+1}=(f_{s}=\text{struck}, - , f_{h}=\text{struck}, - , - ), - , - , - , - )\]

John is the observed performing automaton \(a\). Struck is the transformation response \(f_{i+1}\) of \(a\), and since it is observable through the receptors for sight and hearing, it takes the first and third position within the parentheses of the five elements. Table is the environment \(e_{i}\). The symbol, -, indicates an unknown value for each of the remaining elements of the 16-tuple. The time and processor graphs can be determined in a similar manner to complete the knowledge structure for the sentence.

The knowledge of this first sentence relates to procedural-type knowledge. Procedural knowledge is defined as the knowledge of knowing how. Algorithms for extracting and storing the meanings of sentences of this first class are discussed in Section 4.2.

Consider for a second example the knowledge structure for the fourth sentence:

4. “The table \((E_{i}')\) has a glass top \((e_{i})\).”

Its meaning can be stored in the 16-tuple for the environment graph in the following manner:

\[(e_{i}=(\text{glass top}, - , - , \text{table}), - , - , - , - , - , - , - , - , - , - , - , - , - , - )\]

The four elements contained in the parentheses (glass top, -, -, table), identify the environment \(e_{i}\). This four-element parentheses is discussed in Section 4.1 and has the general form, \(e_{i}=(e_{i}, f_{i}(e_{i}), g_{i}(e_{i}), E_{i}')\). \(e_{i}\) is a part of, or element of \(E_{i}'\). \(f_{i}(e_{i})\) pertains to a functional use of the environment; e.g., \(f_{i}(e_{i})=\text{reading}\), if the table with the glass top is used as a reading table. \(g_{i}(e_{i})\) describes the physical appearance of the environment; e.g., \(g_{i}(e_{i})=\text{round}\), if the table with the glass top is round. As before, the symbol, -, indicates an unknown value for each of the remaining elements of the tuple.

The knowledge of this fourth class of sentence is commonly classified as declarative. Declarative knowledge is defined as the knowledge of knowing that and is acquired suddenly.

The above discussion pertained to knowledge structures for sentences describing single automata. There are also knowledge structures for sentences describing systems of interactive automata. The automata interact by sharing environments, where a recorded response of one automaton is later interpreted as a stimulus by another automaton of the system. Sentences describing interactive automata often contain words of give, receive and sell, buy. The knowledge structures for these sentences are discussed in Chapter 6.
An example sentence containing *give* will be analyzed here, and its meaning will be extracted and stored as a knowledge structure. The example sentence is

“John gives Mary a book.”

Two knowledge structures sharing a single point are required to store the meaning of the sentence. The shared point represents an environment. A first structure consisting of two points and an adjacent line, is required for *John*, the giver of the book, and a second, also consisting of two points and an adjacent line, is required by *Mary*, the receiver of the book. The two graph tuples for these two knowledge structures are shown in Figure 1.2. The two corresponding knowledge structures are joined at the common environment point $E'_{k+1} = E'_{i+1}$. The resulting structure is also shown in Figure 1.2. The part of the knowledge structure with the solid lines pertains to *John* and the part with the broken lines pertains to *Mary*. A shared environment is required if a transfer of the book from *John* to *Mary* is to take place. When the path of *Mary* is followed through the structure, an equivalent sentence is generated.

“Mary receives a book from John.”

Figure 1.2 Environment graph representation for the knowledge structure of the sentence: “John gives a book to Mary.”
The knowledge contained in a sequence of sentences describing a sequence of observations of performing automata, or histories of automata, is represented by a sequence, point, line, point, line, ---, point in a graph representing that knowledge. Such a sequence is called an effective operation path. A similar type of sequence for a knowledge structure may be determined and expressed as a sequence of sentences as in storytelling or as in the conveyance of plans for achieving a goal. Derivations of disciplines describing effective operation paths in general automaton graphs and graphs of general systems of interactive automata are presented in Appendices A and B.

1.2.3 Associating Knowledge

The association of current knowledge and previous knowledge is accomplished by combining separate knowledge structures; i.e., by establishing connected graphs. This eliminates a duplication of structures and facilitates operating on the structures. An algorithm for performing associations is discussed in Section 4.3.

Now return to the first pair of example sentences of Section 1.1 and see how their meanings are distinguishable when association of knowledge is employed. The pair of sentences is

“John struck the table with a hammer.”
“John struck the table with a glass top.”

The sentences are syntactically identical but they have different meanings. Assume that some time before the pair is received by an automaton for interpretation, the following sentence was received, and its meaning was stored as a knowledge structure:

“The table \( (E'_i) \) has a glass top \( (e_i) \).”

\[
(e_i = (\text{glass top, -,-, table}), -,-,-,-,-,-,-,-,-,-, (,-,-,,-,-),(,-,-,,-,,-))
\]

This knowledge structure, now in the memory of the automaton, is an aid in the interpretation of the pair of sentences. The knowledge structure for the sentences become

“John \( (a) \) struck \( (f_{i+1}) \) the table \( (e_i) \) with a hammer \( (z'') \).”

\[
(e_i = \text{table, -,-,-,-,-, z''=hammer, - a=John, - f_{i+1} = (\text{struck, -, struck, -,-}), -,-,-,-,-})
\]

“John \( (a) \) struck \( (f_{i+1}) \) the table \( (E'_i) \) with a glass top \( (e_i) \).”
Thus, the true meanings of the pair of sentences are stored as knowledge structures and are readily distinguishable. That is, when this pair of sentences is received by the automaton, it is clear from previously stored knowledge that the glass top is a part of the table and the hammer is an auxiliary effector of John.

There is also a need to refer to past knowledge in establishing the meanings of the second pair of example sentences of Section 1.1. The sentences are

“The deer came out of the wood.”
“The worm came out of the wood.”

For these sentences, wood has different meanings. Assume that some time before the pair is received by an automaton for interpretation, the following sentences were received, and their meanings were stored as knowledge structures:

“A deer is greater than three feet in height.”

\((e_i=(\text{deer}, -, >3 \text{ ft. high}, -), -, (-, -, -, -, -))\)

“A worm is less than one foot long.”

\((e_i=(\text{worm}, -, <1 \text{ ft. long}, -), -, (-, -, -, -, -))\)

“A wood is greater than 100 feet square.”

\((e_i=(\text{wood}, -, >100 \text{ ft. sq.}, -), -, (-, -, -, -, -))\)

“A wood is less than three feet in diameter.”

\((e_i=(\text{wood}, -, >3 \text{ ft. diam.}, -), -, (-, -, -, -, -))\)

For the first two sentences, \(a\) has a general form that is similar to the general form of \(e_i\) for the last two sentences. The environment was previously defined as \(e_i=(e_i, f_i, (e_i), g_i, (e_i), E_i)\). The automaton is defined as \(a=(a, f (a), g (a), A')\).

These knowledge structures, now in memory, aid in the interpretation of the pair of sentences. The sentences and their knowledge structures are

“The deer \((a)\) came out \((f_{i+1})\) of the wood \((e_i)\).”
Knowing about automata as a subset of world knowledge

The worm (a) came out (e_{i+1}) of the wood (e_i).

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.

It will now be shown how the model for knowledge structures based on general automata accommodates the duplication of the example performances described in Section 1.1. The first four require the procedures for associating knowledge and will be discussed here. The last two will be discussed in the next section. The environment graph and time graph will be used as knowledge structures, and knowledge associations will take place on these structures.

Recall the first performance.

1. An observer sees Mrs. Bee shop at Supermarket Dee one day, the next day sees Mrs. Cee shop at the same supermarket, and then says,

"Mrs. Bee and Mrs. Cee shop at Supermarket Dee."

The proper environment (wood) was selected in each case from past knowledge of the size of the involved automata, deer and worm. You, the reader of Section 1.1, very likely had already selected the proper environments for the automata from your past knowledge.
“Mrs. Bee and Mrs. Cee shop at Supermarket Dee.”

This type of knowledge association can appropriately be called *space association*.

Consider the second performance of Section 1.1.

2. An observer one day reads that during the Renaissance period, the people of Venice built the Library of St. Mark, and years later reads that during the Renaissance period, the Romans built the Church of St. Peter, and then says,

> “During the Renaissance period, the people of Venice built the Library of St. Mark, and the Romans built the Church of St. Peter.”

Knowledge association takes place in a manner similar to that of the first performance except the graphs as knowledge structures are time graphs instead of environment graphs. The observer first establishes a knowledge structure, in the form of a time graph, of the *people of Venice* building the **Library of St. Mark during the Renaissance period**, and the tuple is

\[
(t_i = \text{Renaissance Period}, e_i = \text{Venice}, \cdot, \cdot, \cdot, \cdot, \cdot, a = \text{people of Venice},
\]

\[
b_{i+1} = \text{Library of St. Mark}, f_{i+1} = (\text{build}, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot)
\]

Years later, the observer establishes a second Knowledge Structure of the **Romans** building the **Church of St. Peter during the Renaissance Period**, and the corresponding graph tuple is

\[
(t_i = \text{Renaissance period}, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot, a = \text{Romans},
\]

\[
b_{i+1} = \text{Church of St. Peter}, f_{i+1} = (\text{build}, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot)
\]

The *Renaissance period* represents time for both the building of the **Library of St. Mark** and the **Church of St. Peter**, so that the two separate time graphs have a common point, \(t_i = \text{Renaissance period}\). The common point gives a single connected time graph, and the knowledge association process is complete. Operations on the connected structure leads immediately to the sentence

> “During the Renaissance period, the people of Venice built the Library of St. Mark, and the Romans built the Church of St. Peter.”

This type of knowledge association can appropriately be called *time association*.

Consider now how the model for knowledge structures accommodates the duplication of the example Performance 3 of Section 1.1. Performance 3 is
3. A person, at one time, hears the sentence

“John ate a nut.”

and at a later time says to a person of little experience,

“John ate a nut. He put it into his mouth. He chewed the nut with his teeth and swallowed it.”

The person who elaborated on eating may be said to have previously observed or read about a number of people eating a variety of foods at different times and to have stored the detailed knowledge about eating in the form of graph tuples. Much of the knowledge contained in the graph tuples can be described by the following sentences:

1. “a ate ei.”
2. “a put ei into a’s mouth.”
3. “a chewed the ei with a’s teeth.”
4. “a swallowed ei.”

a is a variable and is any one of the people who was observed eating food. ei is also a variable and is the corresponding food that was observed being eaten.

Figure 1.3 shows the four graph tuples for the above sentences combined into a single environment graph, and the knowledge association process is complete. The numbers on the graph correspond to the numbers identifying the above sentences. The graph is described in greater detail in Section 4.3.

When the sentence, “John ate a nut,” is received, the variable a is given the word value of John, and ei is given the word value of nut. The person is then able to operate on the stored knowledge structure and to respond with the details

“John ate a nut. He put the nut into his mouth. He chewed the nut with his teeth and swallowed it.”

Figure 1.3 Four graph tuples for eating combined into a single environment graph.
It was shown that the model for knowledge structures accommodates the duplication of the example Performances 1, 2, and 3 through the use of knowledge association, and knowledge association was accomplished by combining points of graphs. The model for knowledge structures also accommodates the duplication of Performance 4 of Section 1.1 through the use of knowledge association, but knowledge association, in this case, is accomplished by combining parallel lines instead of points. Performance 4 is:

4. An observer sees and almost simultaneously hears the oral sound of an animal one day, hears and almost simultaneously smells (but does not see) that same animal at some later time, and still later, only smells that animal, and then draws a picture of that animal in answer to the question “What is that?”

The basis for combining parallel lines into single lines references the functions of the graph model. There is a graph function for each of the graphs, Processor, environment, and time graph, and the ordering of terms identifying two points and an adjacent line is not arbitrary but is a grouping of terms for a domain point and of terms for its image. Suppose an observation was made of a performing automaton at some moment. Assume the observation gives a domain point consisting of a group of terms \( (a, b, c, d) \) and a corresponding image point consisting of a group of terms \( (e, -, g, -, i) \). The dashes, -, indicate unknown (unobserved) values. Suppose at some moment later, an observation was made giving a domain point consisting of the same group of terms \( (a, b, c, d) \). Then, if a corresponding image point consisting of terms \( (-, f, -, h, i) \) was also observed, the two graphs (each of two points and an adjacent line) could be combined giving a single graph whose domain point is \( (a, b, c, d) \) and whose image point is now \( (e, f, g, h, i) \). This must be true based on the single-valueness of the function. This type of knowledge association and how it relates to similar performances are discussed in greater detail in Section 4.4.

1.2.4 Establishing Conclusions and Inferences

Conclusions and inferences are based on knowledge previously acquired, and knowledge association, discussed in the previous section, becomes paramount. Example Performances 5 and 6 described in Section 1.1 yield responses that are related to conclusions and inferences. It will be shown how the model for knowledge structures based on general automata accommodates the duplication of these example Performances 5 and 6.

Performance 5 is
5. A person hears the statement

“Copper is a metal.”

and then responds with the statement

“Copper is malleable.”

without having previously received that responding statement.

The two conversational statements are recognized as parts of the valid argument

“All metals are malleable.
Copper is a metal.
Therefore, copper is malleable.”

The first two sentences of the argument are premises, and the last sentence is the conclusion.

Relate Performance 5 to operations on knowledge structures. The first premise

“All metals are malleable.”

can be considered to have been previously stored as a Knowledge Structure described by the tuple

\[ (e_1 = (\text{metal, malleable, -}) \) \]

When the second premise

“Copper is a metal.”

is received, it is stored as a Knowledge Structure described by the tuple

\[ (e_2 = (\text{copper, -}) \) \]

The association of (metal, malleable, -) and (copper, -) for identifying the environment \( e_1 \) yields

\[ e_1 = ((\text{copper, -}), \text{metal, malleable, -}) \]

and the graph tuple that contains the meaning of both premises is
Now, suppose an operation is performed to insert malleable in the position following the word copper in the above tuple for the two premises. Then the following tuple is obtained:

\[ (e_i = ((\text{copper}, \text{malleable}, -), \text{metal}, \text{malleable}, -, -, -, -, -, -), \) \\
\]

But this is the knowledge structure for the complete argument, and an operation on the structure yields the conclusion

“Copper is malleable.”

which is the response in Performance 5. These operations on the knowledge structures are discussed in greater detail in Section 4.4.

The last performance, Performance 6 in Section 1.1, yields a response that involves an inference. To infer suggests the arriving at a decision or opinion by reasoning from known facts or evidence. It will now be shown how the model for knowledge structures accommodates the duplication of Performance 6, which is

6. A person hears the statement

“Mary put the nut into her mouth.”

and then responds with the statement

“Mary probably chewed the nut.”

Assume the person had obtained knowledge about eating from observing or reading about individuals over a period of time eating various foods. Some of the knowledge the person obtained about eating can be described by the sentences

1. “a ate e_i,”
2. “a put e_i into a’s mouth.”
3. “a chewed e_i.”
4. “a swallowed e_i.”

where a and e_i are variables. Following this storage of knowledge, the person of Performance 6 hears the sentence