RICE UNIVERSITY

COGAP-42: Game Playing and Learning
In a Non-Board, Imperfect Information Game

by

Mary Sandra Carberry

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

Thesis Director's signature:

Edward Alvin Feustel

Houston, Texas

May, 1970
ACKNOWLEDGMENTS

The author particularly wishes to thank Professor Edward A. Feustel for his generous expenditure of time and his valuable advice and guidance in the development of this thesis. She also appreciates the helpful suggestions from Professors Schum and Sibert. She wishes to thank the many people who contributed to this project by analyzing and bidding "42" hands. And most of all, she wishes to thank her husband John for his help, suggestions, and encouragement throughout.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td><strong>1. MECHANICAL GAME PLAYING AND LEARNING</strong></td>
<td>5</td>
</tr>
<tr>
<td>Why Study Games?</td>
<td></td>
</tr>
<tr>
<td>Game Playing Programs: Historical Survey</td>
<td></td>
</tr>
<tr>
<td>The Use of Learning</td>
<td></td>
</tr>
<tr>
<td>Types of Learning</td>
<td></td>
</tr>
<tr>
<td>Learning Programs: Historical Survey</td>
<td></td>
</tr>
<tr>
<td><strong>2. &quot;42&quot; AS A GAME PLAYING PROGRAM</strong></td>
<td>37</td>
</tr>
<tr>
<td>Why Study &quot;42&quot; as an Experiment in Game Playing</td>
<td></td>
</tr>
<tr>
<td>COGAP-42: The &quot;42&quot; Bidding Machine</td>
<td></td>
</tr>
<tr>
<td>Signature Table Evaluation</td>
<td></td>
</tr>
<tr>
<td>Sample Signature Table</td>
<td></td>
</tr>
<tr>
<td>Basic COGAP-42 Evaluation and Bidding Procedures</td>
<td></td>
</tr>
<tr>
<td>Signature Table Features: Non-Linearity and Goal Orientation</td>
<td></td>
</tr>
<tr>
<td>Imperfect Information and Team Play</td>
<td></td>
</tr>
<tr>
<td>COGAP-42 and Types of Bidding</td>
<td></td>
</tr>
<tr>
<td>Additional Remarks</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td></td>
</tr>
</tbody>
</table>
3. "42" AS AN EXPERIMENT IN LEARNING

   Why Study "42" as an Experiment in Learning

   Introduction to COGAP-42 Learning

   Generalized Learning

   Checker Player II Signature Table Learning

   Introduction to COGAP-42 Generalized Learning

   COGAP-42 Signature Table Reinforcement:
      Offensive

   COGAP-42 Signature Table Reinforcement:
      Defensive

   COGAP-42 Signature Table Categorization

   COGAP-42 Layered Learning and Smoothing Operations

   Rote Learning

   COGAP-42 Simulation of Human Learning

   COGAP-42 Hand Generation for Learning

   Summary

4. HEURISTICS AND DATA ALLOCATION

   Introduction

   Heuristic Parameters

   COGAP-42 Bidding Structure

   Probability Heuristic

   Summary

   Data Allocation
5. RESULTS.................................................. 131

   Introduction
   Improvement and Convergence
   Layered Learning
   Simulation of Human Bid Determination
   The Probability Heuristic and Range
   Categorization
   Summary
   Future Research

CONCLUSIONS........................................... 151

APPENDIX I - RULES AND PLAY OF "42"............. 154
APPENDIX II - ALPHA-BETA ALGORITHM.............. 162
BIBLIOGRAPHY.......................................... 167
ABSTRACT

COGAP-42: Game Playing and Learning
In a Non-Board, Imperfect Information Game
Mary Sandra Carberry

This thesis investigates a new area of mechanical game playing and learning. The first part surveys previous game playing and learning research. The second and third parts develop a game playing/learning machine. The fourth part presents an inductive definition of heuristic parameter and analyzes the heuristic parameters used in this machine.

This project expands mechanical game playing and learning to a new class of problem. "42" bidding was chosen for investigation because it differs in many respects from previous areas of experimentation. The major differences are: 1) non-board, 2) non-zero-sum, 3) imperfect information, and 4) partnership interaction. These factors required that the machine be designed as a two phase system — hand evaluation and bid determination — with rote and generalized learning acting interdependently. Strategy is introduced via a goal-oriented evaluation and learning procedure. This should more closely parallel human decision making functions and strategical considerations than standard non-goal evaluation and learning. A heuristic is developed for estimating the probabilities necessitated by imperfect
information; other heuristics take into consideration both imperfect information and team play.

The number of possible dominoe distributions and bid sequences prohibit rote learning as the primary learning scheme. The signature table procedure, introduced by A.L. Samuel, is used to effect generalized learning. However, problem complexities and resultant memory requirements prevent the use of common reinforcement methods for generalized learning. This research develops a new weighted reinforcement method for the signature table procedure. A range categorization scheme is developed to effect better categorization of signature tables having non-uniform distributions of signatures. "Layered learning" --- the learning of an entire layer of instances at one level before abstraction to the next level --- is introduced to alleviate the abstraction difficulties in a hierarchial learning structure. This layered learning increases the speed and accuracy of learning.

Complex heuristic parameters are utilized in the "42" machine in order to provide informative and meaningful heuristic descriptions of sub-goals within the goal-oriented evaluation procedures. An inductive definition of heuristic and heuristic parameter is developed to classify such heuristic parameters according to degree of complexity and to afford a comparison of different levels of heuristics.
INTRODUCTION

Conventional programming is the programming of algorithms that detail the step-by-step solution or operation of a problem or process. The composition of music, the invention of scientific devices, and most such intelligent behavior cannot be algorithmically described. Limitations on machine size and speed exclude a large percentage of the remaining areas of consideration. For example, Shannon (18) has estimated that over $10^{120}$ move continuations must be examined in a full chess look-ahead procedure. Even if computer size and speed were increased to several times present day standards, construction of a successful chess player would have remained unfeasible.

Research in artificial intelligence has expanded computer applications to include processes requiring intelligent thought and judgment decisions. During the past decade, machines have used mechanical reduction techniques in theorem proving. Heuristic problem solvers, patterned after human thought processes, have successfully attacked difficult problems in logic and mathematics. Game playing programs have defeated above average opponents in chess and checkers. Machines simulated human thought and decision making in such areas as concept formation and trust investment. And machines, like men, have modified their behavior by learning from experience.
This research project expands mechanical game playing and learning to a new class of problems. The game chosen for investigation is "42", a dominoes game similar in many respects to the card game bridge. Four players, forming two partnerships, are required; each player is dealt a hand of seven dominoes. After analyzing his hand, each player is allowed one and only one bid; the player making the highest bid obtains the contract for his side. This team then attempts during the course of play to take enough tricks and special point dominoes to make the contract bid. The detailed rules of "42" and sample play are presented in Appendix I.

"42" bidding was chosen as the subject of learning research both for its simplicity and its complexity. Each player receives seven dominoes compared with thirteen cards in bridge; only one round of bidding is allowed instead of several progressively higher bids. However, certain dominoes have special point value and must be given special consideration in any bidding. These factors lessen memory requirements considerably yet maintain a complexity that prevents algorithmic bid determination.

This research seeks to investigate several new phases of game playing and learning. "42" is a non-board, non-zero-sum game requiring cooperative team play. Previous research games have maintained perfect information—there is no chance element and each player possesses all information necessary to determine the best line of play. "42"
has imperfect information: each player sees only his own hand—the dominoes held by partner and opponents are unknown and can only be probabilistically estimated. These four new factors—non-board, non-zero-sum, partnership interaction, and imperfect information—must be adequately accounted for in a "42" machine.

The "42" bidder is designed as a two phase system—hand evaluation and bid determination—with rote and generalized learning acting interdependently. Strategy is introduced via a goal-oriented evaluation and learning procedure. This should more closely parallel human decision making functions and strategical considerations than standard non-goal evaluation and learning. A heuristic is developed for estimating the probabilities necessitated by imperfect information; the goodness of this heuristic as a measure of the true probability will severely affect success of this machine.

The number of possible dominoe distributions and bid sequences prohibit rote learning as the primary learning scheme. However, problem complexities and resultant memory requirements also prevent the use of common reinforcement methods for generalized learning. Thus, one of the primary objectives of this research is the modification of non-linear generalized learning schemes and reinforcement techniques for application to a wider range of problems, including "42".

In order to increase the learning rate, layered learning is defined and implemented. This layered learning
technique alleviates hierarchial abstraction difficulties during generalized learning. Although necessitated by this particular project's time limitations, this technique will assume greater importance as problem complexity increases.

Complex heuristic parameters are utilized in the "42" machine in order to provide informative and meaningful heuristic descriptions of sub-goals within the goal-oriented evaluation procedures. An inductive definition of heuristic and heuristic parameter is developed to classify such heuristic parameters according to degree of complexity and to afford a comparison of different levels of heuristics.

Chapter 1 surveys research in mechanical game playing and learning. Chapters 2 and 3 respectively describe the game playing and learning aspects of the "42" project. Chapter 4 presents the inductive definition of heuristic, classifies the "42" heuristic parameters, and describes in detail the probability heuristic; this chapter also discusses data structures utilized in the "42" machine. Chapter 5 presents and analyzes results of this project. Within the appendices will be found a detailed description of the "42" rules and play and the alpha-beta algorithm.
Chapter 1

MECHANICAL GAME PLAYING AND LEARNING

Why Study Games?

Game playing has intrigued many researchers in artificial intelligence. The concern is not with some particular game itself but with the characteristics of certain games that make them good tools for research in mechanical problem solving.

A true game is non-determinate in the sense that no easily utilized algorithm exists guaranteeing a win. Therefore intelligent decision making processes are essential for winning play. In essence, a game is a problem. The problem framework is the set of rules governing play of the game. The problem itself may be stated: "within the problem framework, devise a winning line of play."

Actually, a game is several problems in one, since each move requires solution of the subproblem: "within the problem framework, determine the best move for the given game situation." The problem is correctly solved by the player winning the game.

Two important facets of problem solving research are testing and evaluation of the mechanical problem solver. That a machine can solve one particular problem is no proof that it possesses good decision making ability. A game provides a framework within which an extremely large number of different formulations of the game problem may be posed.
The machine solution of each problem may be compared with various human solutions or with a store of "expert" play.

These qualities — the indeterminate nature of a game, its similarity to a problem environment, and its ease of testing and evaluation — have resulted in construction of a number of successful game playing programs.

Game Playing Programs: Historical Survey

Game playing programs have undergone a long period of development. The notion of a mechanical chess player has intrigued man for centuries. The growth of this chess machine will be presented in some detail for several reasons:

1) The step-by-step development of mechanical chess playing is well documented in the computer literature.
2) This development illustrates the evolution of intelligent machines from a mere concept to a realized entity.
3) Many of the basic concepts of the chess playing programs appear in other game playing machines and are therefore worthy of careful consideration.

In 1949 Claude Shannon (18) presented the first major paper on chess playing machines. Shannon suggested that an evaluation function be used to measure the goodness of a particular board position. This evaluation function was a polynomial of the form:

\[ f(P) = A_1(T_1) + A_2(T_2) + \ldots + A_n(T_n) \]
where \( P \) is a board position; \( t_i \) is the value of a selected chess parameter; and \( a_i \) is an experimentally determined weight. Chess is a zero-sum game; that is, the goodness of a particular board position to one player is necessarily a measure of the badness of this situation to his opponent. Accordingly, each parameter \( t_i \) is the value of a measured chess quantity for player \( L \) minus the value of this quantity for opponent \( L' \). \( f(P) \) assumes a positive or negative value as board position \( P \) favors player \( L \) or \( L' \) respectively.

Shannon suggested that such a polynomial function be used to evaluate final board positions in a look-ahead procedure, and that minimaxing then be applied to determine the correct initial move. In such a procedure, the entire move tree is expanded several levels of play and the resultant board positions are assigned scores via the evaluation polynomial. Player \( L \) then chooses the line of play that maximizes the value of the final board position, always assuming that opponent \( L' \) chooses moves minimizing it.

As an example of minimaxing, consider the move tree of Figure 1, expanded two moves for player \( L \) and one move for opponent \( L' \). The final board positions are labelled with their \( f(P) \) values. At node A, player \( L \) will choose to maximize his advantage and will therefore select the move leading to a board value of 3. Board position A therefore assumes the value 3. Similarly, nodes B, C, and D receive values 2, -3, and 8 respectively. At node E,
opponent L' will move to maximize his advantage or, equivalently, to minimize the final $f(P)$. L' therefore chooses to move to node B since this limits the final board value to 2. Board position E therefore assumes value 2 and F assumes value -3. Similarly player L will choose to move from node G to node E, guaranteeing that the final board value must be at least 2. Thus the correct line of play from node G, given that each player attempts to maximize his advantage and assumes his opponent will do the same, is E-B-to a final board position of value 2.

If one could look-ahead to any depth, one could consider all lines of play until termination of the game,
and minimax to determine whether each initial move for player L leads to a win, lose, or draw. However, Shannon estimates that $10^{120}$ move continuations must be examined—an impossible situation. Therefore the model looks ahead to a specific depth, scores these final board positions via an appropriate evaluation polynomial, and minimaxes to determine the prescribed initial move.

Shannon's evaluation function applied during a look-ahead procedure formed the basis for future work in game playing. However, Shannon also introduced several other notions which have subsequently been incorporated into modern chess playing machines. These included:

1) expansion of the move tree until attaining a relatively "quiescent" state, since board evaluation in the midst of a series of captures is meaningless.

2) the discarding of pointless move continuations.

3) the use of three evaluation polynomials since the opening, middle, and end games each have peculiar principles of good play.

Although Shannon discussed the programming of such a chess player, the first chess playing program was constructed by A.M. Turing (19) in 1951. Turing adopted Shannon's evaluation function and look-ahead procedure. Turing's evaluation function consisted primarily of one parameter reflecting material balance, the difference in pieces on the chessboard. Other parameters entered into the evaluation function only if forced to choose between two moves equal in material. Turing expanded the move tree
until reaching a quiescent state or "dead" position at each node; he then scored each board position and minimaxed to determine the initial move.

Turing resorted to hand simulation of his chess player. Although very weak in ability, it led to development of the first actual chess playing machine by a group of Los Alamos researchers.

The Los Alamos program (12), designed by J. Kister, P. Stein, S. Ulam, M. Walden, and M. Wells, played a reduced game of chess on a 6x6 board, omitting bishops and four pawns. The evaluation polynomial measures both material advantage and a second factor, mobility. The move tree was expanded to a depth of two moves per player, the final board positions evaluated, and minimaxing applied to determine the initial move.

This machine showed promise by defeating rank beginners but it lost to experienced players even if given a several piece advantage.

The next significant step was accomplished by Alex Bernstein (14) in 1957. He developed a full chess playing machine on the IBM 704. Although its quality of play was not necessarily superior to the Los Alamos program, this chess player is significant in that it decreased the number of moves examined by increasing program complexity.

Bernstein incorporated into his machine a series of plausible move generators, each of which suggested a move related to a particular chess strategy. At each node of the move tree, these generators were activated to determine the
seven move continuations to be examined. This was done to a depth of four plys (two moves per player). At this point, Bernstein applied his evaluation function to the resultant $7^4$ final board positions and minimaxed to determine the initial move. Considering that at each node there are approximately 30 legal moves, only a fraction of the possible move continuations were examined.

The evaluation function differed somewhat from that suggested by Shannon. Bernstein used a ratio of two polynomials, one for each player. Each polynomial measured material, mobility, King defense, and area control.

Bernstein's chess player pointed the way to greater depth of analysis during look-ahead — not by a fast machine but by increased selectivity and complexity of the chess program itself. Thus chess machines began to incorporate heuristic methods — rules of thumb or strategy to limit and guide the search for a solution.

Newell, Shaw, and Simon (14) presented their chess player in 1958. Although considerably more sophisticated than previous chess machines, the NSS program contained selected features of its predecessors.

To determine the best chess move, the NSS machine activated a set of goal specification routines, obtaining a priority ordered goal list relevant to the particular board situation. This list contained such goals as King safety, material balance, or other recognized chess tactics. Associated with each goal was a plausible move generator
which proposed moves to achieve its allied goal.

The goals were considered sequentially on the priority ordered goal list. Once the associated move generator had proposed an initial move, the analysis move generators assumed control. Each analysis move generator determined if the new board position was "dead" with respect to its associated goal; if not, a set of plausible moves were generated and the analysis move generators were reactivated for the new board positions. Once a set of positions that were dead relative to all goals on the goal list was obtained, static evaluation occurred. Each final board position was scored as a vector whose components were the values of the given position with respect to the corresponding goals on the ordered goal list.

The ordering of the vectors was lexicographic:

\[ V(A_1, A_2, \ldots, A_n) \succ V'(A_1', A_2', \ldots, A_n') \]

if and only if

\[ A_i \succ A_i' \] for some \( i \)

and \( A_j \preceq A_j' \) for any \( j < i \)

The final board position scores thus were completely ordered; minimaxing determined the vector value of the initial move suggested by the goal-oriented plausible move generator.

Instead of activating each plausible move generator in turn and comparing the scores of the various suggested moves, the NSS machine had a pre-determined acceptance level. The moves were considered in order of their goal
priority, and the first move meeting this acceptance level was chosen. If no move fulfilled the acceptance criteria, the best move generated was made.

This Newell-Shaw-Simon chess program was far more complicated and heuristically oriented than any of its predecessors. It appeared to achieve amateur status and indicated that the search for better heuristics would lead the way to a more competitive chess machine.

The most successful of the chess machines was developed by Richard Greenblatt (7) on the PDP-6 computer of Project MAC at M.I.T. The ability of the Greenblatt program was midway between that of the overall average chess player and that of the average U.S. tournament player. Under the name of Mac Hack Six, it has won the amateur non-master Class D trophy.

To determine the best move, the Greenblatt program activated a set of plausible move generators. These generators used about fifty different heuristics to propose a set of moves ordered according to their plausibility. For each resulting board position, the plausible move generators were reactivated until the move tree had been expanded to a specified depth. The position evaluator then assumed control. The presence of plausible captures or a feedover condition caused the position evaluator to reactivate the plausible move generator and further expand the search tree. (The feedover condition related to certain en prise chess situations and will not be discussed here.)
Once the position evaluator determined that a set of static board positions had been obtained, the static board evaluator scored the resultant final boards. The evaluation function was a polynomial consisting of five terms: material balance, piece ratio change, pawn structure, King safety, and center control. Minimaxing determined the best initial move.

Additional features included: use of the alpha-beta tree pruning algorithm (described in Appendix II) to reduce the search tree; "hash" coding to reduce the search by preventing a second consideration of a given position; generation of a secondary search whenever a new best move was found at the top level; and inclusion of a store of over 5000 book moves for opening games.

The Greenblatt program has achieved the most success as a mechanical chess player. It is interesting to note that Shannon's evaluation polynomial and look-ahead procedure, Turing's dead position, Bernstein's plausible move generator, and Newell-Shaw-Simon's greater use of heuristics were all adopted and expanded in developing this superior chess machine.

Although many game playing programs have been constructed, the following discussion will focus on major works in several different areas.

Kalah is an ancient Middle Eastern game. Figure 1-2 depicts a Kalah board. Each player has a specified number of pebbles, usually six per pit. Each of the two players in
turn removes the pebbles from one of his pits and distributes them one-by-one counter-clockwise around the board, omitting the opponents Kalah. Any pebble landing in the player's own Kalah remains there; if the last pebble is placed in one of the player's own empty pits, the pit directly opposite on the opponent's side loses its pebbles to the player's Kalah. The object of the game is to accumulate the most pebbles in one's own Kalah.

A player receives another turn if his last distributed pebble lands in his Kalah. The game ends when one player's pits are empty. The opponent then moves all his remaining pebbles to his Kalah and the number of pebbles is tallied to determine the winner.

A Kalah machine was developed by A.G. Bell. To determine the best move, this mechanical Kalah player expanded the move tree to a depth of three plys. This expansion took into consideration move continuations due to receiving one or more extra turns; a tree was formed whose
branches were ordered according to the number of moves occurring between nodes.

The final board positions were scored by an evaluation function that measured the difference in pebbles contained in the player's Kalah and the opponent's Kalah. The player sought to maximize this difference, the opponent to minimize it. Each final board position was matched against a store of winning-losing positions. If the board position was found to be a part of this store, it was evaluated appropriately as a win or as a loss. Minimaxing determined the best initial move. Application of the alpha-beta algorithm during minimaxing and the ordered nature of the tree acted to reduce the magnitude of the search.

This Kalah player was implemented on the S.R.C. Atlas computer at Chilton, Berkshire. It played a superior game against both weak and strong opponents.

The game of Go-Moku, although easy to describe, is difficult to play well. Each player in turn places a stone of his color on a lattice point of a 19x19 mesh board. The object of the game is to complete a horizontal, vertical, or diagonal string of one's own stones.

E.W. Elcock and A.M. Murray (3) undertook construction of a Go-Moku program primarily as an experiment in learning. The learning aspect will not be discussed here. The program contained a store of board descriptions, called subgoals. Each subgoal was a description of a class of possible moves; this description had component sub-
descriptions that described particular features of board patterns. The sub-goals were ordered on the goal list by the number of moves away from a win.

To determine the best Go-Moku move, a list of legal moves was first generated. The Go-Moku player searched its priority ordered goal list to find the highest sub-goal that described one of its available moves. The machine chose this move unless the opponent had a higher pertinent subgoal; then the player was forced to make its highest defensive subgoal move on the lattice point related to the opponent's subgoal. A default procedure was activated if the subgoal search produced no available move. This routine selected a move by other methods, sometimes via a random move generator.

The Go-Moku program introduced pattern recognition techniques into game-playing programs. Albert Zobrist (20) constructed a Go machine incorporating more sophisticated heuristics for pattern recognition.

The game of Go is deemed by some to be of greater complexity than chess. Each player in turn places a stone of his color on one of the lattice points of a 19x19 mesh board. A string of stones of one color may form a chain of horizontal or vertical segments. Empty lattice points horizontally or vertically adjacent to this chain are called breathing spaces. A chain with no breathing spaces is captured by the opponent and removed from the board. A player may pass, but passes by both players in succession
ends the game. Each player then sums pieces he has captured, empty vertices, and the opponent's board pieces that are completely surrounded by his color to determine the winner. Two additional rules prevent a player from removing all breathing spaces from his chain unless it results in capture of the opponent's pieces and from capturing an opponent's piece immediately after it has caused his capture.

Zobrist's Go machine differed considerably from previously discussed game-playing programs. It was designed as an investigation into human problem solving processes and perceptual abilities. The basic framework of the Go program consisted of representing the Go board in a set of meaningful visual segments. Such perceptual heuristics might include grouping of stones of the same color or division of the board into segments over which particular stones wield influence.

To determine the best Go move, the visual organization heuristic produced a $19 \times 19$ grid representation of the perceptual patterns contained on the given board. The internal representation routine constructed from this visual pattern organization seven $19 \times 19$ grids, which measured such features as segmentation, number of breathing spaces in a chain, and other pertinent Go quantities.

The move determination routines possessed a set of "templates" or familiar piece configurations. These templates were relatively general in specification and
could be applied after rotation or reflection of the game board. Sixty-five of the templates indicated a desired move upon matching a portion of the internal board representation. All legal moves had an initial score of zero. A matched template caused its associated weight to be added to the score of the indicated move.

Twenty other templates activated a look-ahead procedure in the general area to which the template was matched. Look-ahead analysis extended four plys; two move trees were formed --- one each for black and white as the first to move. The tree could be expanded further under certain conditions. From this analysis, the program determined if the player was forced to move to avoid capture; if so, 4000 was added to the scores of appropriate moves in this area. After application of all the templates, the Go machine selected the move with the largest score, passing if no move achieved a score greater than 100.

This program has drawn considerable interest not for its ability, which is still weak, but for its sophisticated use of human pattern recognition and visual organization heuristics.

A.L. Samuel (16,17) has devoted several years to investigating learning in the game of checkers. Although the primary emphasis in his work was on aspects of learning, a by-product was the development of a machine checker player of exceptional ability.

Two major checker programs were developed by Samuel.
The first (16) used similar techniques to those of the chess playing programs. The second (17) is a major advance in allowing non-linear interaction of the checkers parameters.

Checker Player I (16) expanded the move tree to a depth of three plys, then expanded selected portions of the tree considerably further upon detecting specific jump conditions. The final board positions were scored by an evaluation polynomial and minimaxing determined the best move.

It should be noted that in all game-playing programs discussed thus far, the evaluation function allowed only for linear interaction of the game parameters. Thus, the machine could not account for cases in which parameter A was of immense importance if parameter B had a large value, but of negligible interest otherwise. Samuel found this to be a major drawback of the conventional evaluation function and attempted to correct this by introducing parameters that were in reality the conjunction of two checkers quantities. However time and space considerations required that the number of combinational terms be severely limited.

Checker Player II (17) replaced the linear evaluation function with a non-linear "signature table" procedure. This procedure will be described here and more fully in Chapters 2 and 3; it forms the evaluation function for the author's own "42" project.
A set of parameters is divided into M subsets; the subsets need not contain an equal number of parameters and overlapping of parameters is permitted. Each subset is a signature type. If the subset contains P parameters and each parameter may take on V values, then there are P*V possible interactions within this signature type. Each of these P*V arguments is referred to as a signature.

Given a board representation resulting from a prospective move, the signature for each of the signature types is computed. This signature designates an entry in the table associated with that signature type. From this entry, one obtains a weighting that reflects the value
of this particular signature.

A signature table actually consists of a hierarchial arrangement of sub-signature tables. The values obtained from entries into signature tables of one level may again be combined to form entries into higher level tables. A signature table of three levels with twelve initial parameters is shown in Figure 1-3. Entry into the highest level table produces the final score for the given board representation.

The decision as to which first level parameters should interact to form a signature type and which sub-signature tables at higher levels should be grouped together must be made experimentally. Samuel attempted to group parameters and sub-signature tables so as to reflect their assumed interdependencies.

Checker Player II contained six complete three-level signature tables, one for each of six phases of the checker game. To determine the best move, the move tree was expanded to a depth of three plys. Further expansion of selected portions of the tree occurred upon detection of specified jump conditions. The final board positions were evaluated by the signature table procedure; minimaxing determined the best initial move. Incorporated into Checker Player II were several additional routines to reduce the search tree: the alpha-beta algorithm, move plausibility analysis, and certain forward pruning techniques.
The signature table procedure significantly improved the quality of checkers play, indicating that non-linear interaction of parameters must be given due consideration in future game-playing machines.

This concludes the discussion of game playing programs. The historical survey indicates that mechanical game players have evolved from mere conceptual devices to machines that can now successfully implement and extend human judgment and decision processes.

**The Use of Learning**

Learning is the "revision of opinion on the basis of experience". Intelligent beings are born not with knowledge and judgment ability, but with the capacity for acquiring them. Similarly, research in artificial intelligence has demonstrated that machines may be constructed with the capacity for improving their decision making abilities by learning from experience.

As problems increase in complexity, this self-improvement of the machine's decision making processes becomes significantly more important. Given a particular problem, the correct solution may be obtained from experts in the field. However, in many complex situations, it would be extremely difficult, if not impossible, to consider a sufficient number of problems in enough variation and detail to generalize these problem-solutions into a competent machine decision making apparatus. Thus methods must be devised for constructing machine decision making
frameworks that may be self-adjusted and self-altered as the machine's experience dictates, resulting in a complex, efficient decision making apparatus.

Types of Learning

Two types of learning ability, rote and generalized, have been exhibited by recent machines. Rote learning is the cataloguing in memory of specific instance-response or problem-solution pairs. Thus if a particular problem has been encountered previously, the machine need only search its memory for the correct solution.

Rote learning produces several notable results. Solutions to recurrent problems are immediately available within the machine. The time saved by avoiding re-solution of old problems may be used to examine other problems in greater depth than might otherwise be expedient. New problems may reduce to a set of previously learned subproblems and therefore will be immediately solvable at the point of reduction. This occurs frequently in theorem proving and game playing. The more complex the problem under consideration, the more advantageous is this reduction. In many sufficiently complicated cases, the original problem would otherwise remain unsolvable.

Two factors restrict the widespread use of machine rote learning. As the number of problem-solution pairs in the memory store increases, the search time expended quickly reduces the effectiveness of rote learning as a time saving device. More importantly, machine memory has a finite
capacity which in most cases will be far exceeded by the problem-solution pairs and the data space necessary to sufficiently describe them.

Several attempts have been made to lessen these disadvantages. Larger and faster machines have been developed. The set of problem-solutions has been subdivided according to the peculiar characteristics of problem groups to eliminate the need to search the entire memory store. The utilization of this learning then assumes many of the aspects of a problem in pattern recognition. The memory store has been pruned at intervals to eliminate pairs of marginal utility. This has the effect both of reducing the store search time and of increasing the memory available for future problem-solution pairs.

However, as the complexity of problems for machine solution increases, the disadvantages of rote learning become more significant. Rote learning is quite effective for accurately learning small amounts of information, but it alone is not sufficient to enable machines to develop expert decision making abilities.

Generalized learning is the "learning of solutions or methods of solution for general classes of problems". Closely aligned with this type of learning is the concept of reinforcement. To "reinforce" a behavioral instance is to increase the probability of eliciting a given response upon application of a particular set of stimuli. In generalized learning, the machine examines the charac-
teristics of a particular problem and in some way reinforces the correct solution for this character set.

Several methods of "training" such a machine exist. A Trainer may be used. The Trainer either indicates the goodness of the machine's solution or presents the machine with the correct solution. In the former, the machine is left to find the correct solution independently, being told only when that solution is obtained and perhaps whether it is proceeding to steadily better solutions. In the latter case, the machine may proceed immediately to reinforce the correct response, and this invariably leads to more rapid learning. If an indication of both success and failure is available within the problem itself, a trainer is not essential. Game playing and theorem proving often contain easily recognized success-failure stimuli.

In more complex problems, it is necessary for the machine to group the basic character sets and generalize upon these to form a hierarchial learning structure. In such cases where the learning is to be carried to several levels of abstraction, it becomes important that the initial learned instances be general and basic in nature, rather than some specific exceptional case, since these initial instances will form the "building blocks" for future learning. Learning may still converge to a good decision making apparatus, but the rate of learning will be significantly slower if an appropriately graded training sequence is not adhered to. Paralleling the problem of
establishing appropriate training sequences is that of the probability environment. If the problems presented for machine learning do not closely approximate the actual occurrence of such problems, then the machine will not have adjusted itself properly to achieve the best decision making apparatus for its environment. Thus a good training sequence must be both appropriately graded and characteristic of the problem environment.

Throughout this discussion, it has been assumed that the machine somehow found pertinent characteristics for generalizing problems. In all the learning machines designed thus far, the characteristics or parameters have been generated by the programmer; the machine has been allowed to select and discard from this general set, but machine generation of parameters has not been accomplished. This looms as one of the major problems of learning machines and still appears a long way from solution.

Generalized learning overcomes the time and space limitations of rote learning; however it still has its own inherent difficulties. Generalized learning does not guarantee a correct solution even if the identical problem has been previously encountered. Better reinforcement techniques and learning schemes must be developed to increase the probability of correct responses. Machine generation of parameters would be a major step in this direction.

Rote and generalized learning each have peculiar characteristics that suit them to particular types of
problems. In the following, several successful uses of learning in artificial intelligence will be surveyed.

Learning Programs: Historical Survey

Learning programs have been implemented in program design, problem solving, theorem proving, and game playing. This section surveys a few of the major contributions to each area.

R.M. Friedberg (4,5) experimented with a statistical learning machine for program production in 1958. This machine was intended as an initial investigation into very basic learning mechanisms.

A program was defined as any sequence of 64 instructions or data words, chosen from a set of $2^{14}$ possibilities. Initially the trainer would input one or more data words; Herman, the learning machine, was then activated and would execute the program. After execution, the trainer analyzed the designated output locations to determine if the predetermined relationship between the input and output locations was satisfied.

Herman was notified of his success or failure at the appointed task. For each of the 64 program locations, Herman maintained an active and an inactive instruction. If successful for the given trial, Herman incremented the success counter for each of the active instructions --- all success numbers were relatively scaled to maintain a limit on their values. Each time a failure occurred, one instruction of the "program" was examined, and if certain
criteria were satisfied, the active and inactive instructions were interchanged. After every 64th failure (each instruction had now been examined on one failure), an instruction with a very low success number was selected for replacement with a new instruction generated randomly, containing a pre-determined success number.

Herman would make many trials on the same problem, with different input words. The more successful he became, the fewer were the changes to the program instructions; Herman retained instructions associated with success and discarded those associated with failure.

During experiments with this learning machine, only limited success was achieved. However, machines with the success graded instructions did perform better than those without such a mechanism.

Gaku, a simple learning machine, was developed by Aiko Hermann (8, 9, 10) in 1964. The machine contained a number of interdependent mechanisms related to particular aspects of problem solving; the induction unit attempted to relate the machine's past experience to new problems and to obtain from the new problems information to be used in future solutions.

Gaku's task analyzer first characterized the given problem and searched its past experience record for useful information. Characteristics of past problems were compared with those of the new problem by the abstraction routine, and the induction unit used this to predict a general line of attack. The many mechanisms and sub-mechanisms inter-
acted with one another until a final correct solution was obtained. The solved problem was then analyzed and Gaku abstracted from it to reinforce criteria used in the solution. This information was retained for future problem solving.

It would be difficult to describe briefly the method of operation of the learning segment in more detail since its operation was so closely related to the other units of the machine. Gaku was applied to the Tower of Hanoi puzzle and was successful at learning to abstract from 3, 4, and 5 disk cases the general solution to the n-disk case.

D.L. Johnson and A.D.C. Holden (11) developed a learning machine for proving trigonometric theorems. The learning unit of the machine exhibited many characteristics of human learning: use of recently learned material, greater retention of recurrent items, and forgetting.

The model initially possessed five basic trigonometric identities; it was also capable of symbol recognition and algebraic manipulation. Given a theorem for proof, the machine attempted to heuristically standardize and characterize the theorem into a familiar form. Transformations were applied for simplification and equalization. If the theorem was successfully proven, both the theorem and its intermediate steps became a part of the model's learned store of transformations.

Finite memory capacity and limited search time required that only useful transformations be retained and that these transformations be utility-ordered. To accomplish this,
each transformation was assigned a familiarity quotient; upon completion of a theorem proof, the familiarity quotients of all transformations used in that proof were incremented by a fixed amount and the familiarity quotients of unused transformations were decremented. Forgetting occurred periodically when transformations of lowest familiarity were removed. A forgotten transformation that was subsequently re-developed was entered into memory with a higher familiarity quotient than it possessed at its time of removal.

The association and selection mechanisms attempted to analyze the problem in such a way as to minimize the number of transformations considered for application. The model further simulated human learning by selecting transformations of greatest utility, preferring first to simplify the theorem and then to attempt to equate the two sides.

The machine demonstrated the ability to learn and to efficiently utilize knowledge acquired. This was particularly apparent when the machine was presented with a set of problems of increasing difficulty. Especially noteworthy was the similarity between this learning machine's methods of cataloguing, culling, and selecting the acquired information, and those of the human problem solver.

J.R. Quinlan (13) developed a learning unit to efficiently select operators in the Fortran Deductive System. Within the system were a set of operators used to transform the initial problem into the desired final state or solution. The FDS learning system concerns itself with
the problem of ordering the operators for application instead of the problem of selecting the set of potential operators.

The set of operator-problem pairs was divided into categories determined by two evaluations: the estimated difficulty of getting the problem into an acceptable form for application of the operator, and the estimated difference between the problem state and the desired solution state. Each class had an associated weight learned by the system.

Given a problem, FDS selected a set A of operators. This set was ordered according to the weights for each operator-problem class; the operators were applied in turn, and subproblems similarly analyzed, until a solution was found. FDS then analyzed the tree of solution-attempts; if an "incorrect" operator was tried before the "correct" operator on any subproblem, the associated weights for the incorrect and correct operator-problem classes were decreased and increased respectively.

The success of the FDS learning system was largely dependent upon the goodness of its classification scheme; this classification was especially notable in that it was independent of the problem environment and therefore maintained the integrity of FDS as a general purpose system. Experimentation with this operator-problem learning selection scheme produced very good results.

Samuel's mechanical checker player (16,17) was the first truly capable learning machine. It has influenced
most subsequent work. Three variants of learning were considered: rote, generalized linear, and generalized non-linear. The game playing mechanism of the checker program was described earlier; this section will consider only the learning aspects of the experiments.

Given a particular board situation during play, Checker Player I (16) performed an X-ply look-ahead search, during which final board evaluations were obtained from the store if possible, otherwise from the evaluation function. The original board position with its associated backed-up score was then entered into the position store for future reference.

Upon first glance, this rote learning may appear only to save computation time but to have little effect upon the playing ability of the machine. The ply depth of the look-ahead procedure had to be severely limited for reasons of time and memory capacity; yet this ply limitation is the most serious handicap of the checker playing machine. Rote learning allowed the most significant moves to be analyzed in greater depth than would otherwise have been possible. This was accomplished in the following manner. If one of the final board positions obtained during an X-ply look-ahead was evaluated via the position store, its evaluation was already at least an X-ply backed-up score. If minimaxing selected this board evaluation as the original board score, the score had been backed-up at least 2·X plys instead of X plys. Thus, moves which were encountered most
often and which therefore probably were most significant, would have scores which had been analyzed to a depth many times greater than the X- ply limit.

Special memory store, search, and pruning methods were essential for efficient operation of this rote learning scheme. Each encountered board position was standardized to take advantage of all available symmetries. Newly stored positions were merged at intervals onto a memory tape; records from this tape were then recalled into the machine as needed; special ordering of the records and the positions within each record minimized search time. To further eliminate useless search time and to maintain available store space, rarely accessed board positions and board positions of low ply depth within a large tape record were automatically erased from the position store.

Checker Player I exhibited the value of rote learning in a manner and to a degree not previously visualized; however, the standard limitations of rote learning prevented the development of master machine play.

A variant of Checker Player I included generalized learning. Given a large initial set of parameters, Checker Player I assessed its experience to determine which parameters should be included in the evaluation polynomial and their associated weights. Thus the learning was generalized to the heuristics affecting good checker moves.

It has been stated that the greater the ply depth of the look-ahead analysis, the better will be the move
evaluation. This fact was the basis of Checker Player I's generalized learning procedure. During game play, the difference between the present board score as obtained from a new look-ahead procedure and as determined on the previous move was defined as "delta". This delta value was therefore a measure of the error in the evaluation polynomial, a positive (negative) delta indicating positively (negatively) contributing terms should be weighted more heavily.

The signs of the individual terms in the initial evaluation polynomial were compared with the delta sign, causing an adjustment to a correlation coefficient for each term. These correlation coefficients indicated the relationship between each individual term and the goodness of the scoring polynomial; they were then used to determine the sign and relative magnitude of each weight in the newly "learned" evaluation polynomial. When a particular term had had the lowest correlation coefficient over a pre-determined number of board situations, it was removed from the evaluation polynomial and replaced by a new term from the reserve list.

This version of Checker Player I exhibited unusual ability at generalizing its learning experience. However, it did suffer some limitations. One of these, the linear nature of its evaluation polynomial, was attacked in Checker Player II. Samuel attempted to alter this linearity in Checker Player I by introducing binary connected terms into the scoring polynomial; however, the number of such
terms which could be included was severely restricted. Checker Player II (17) overcame these limitations by using a completely new evaluation procedure — the signature table previously described.

The signature learning scheme was implemented with the use of a trainer. "Book" games were played by the machine; given a board situation, the machine reinforced the correct "expert" move. For each signature type occurring in the evaluation of a correct or incorrect move, an associated A or D tally respectively was incremented. At intervals in the learning process, a correlation coefficient for each signature type was computed from these tallies. These correlation coefficients were the final board scores in the case of highest level signature tables; at the lower levels, they were used to determine the entry into the next table level. This was accomplished by dividing the signature types into the desired number of equisized groups on the basis of their correlation coefficients and then assigning all members of a group the same table entry number. This non-linear learning scheme improved significantly upon the linear evaluation method, even when both were allowed to utilize a trainer.

The historical survey indicates that machines indeed do possess the ability to learn from their experience. Although present learning schemes are still primitive and have many difficulties which must be overcome, learning mechanisms will inevitably play a vital role in the development of machines of superior decision making ability.
Chapter 2

"42" AS A GAME PLAYING PROGRAM

Why Study "42" as an Experiment in Game Playing

"42" bidding was selected as the subject of this research because it possesses several characteristics that are new to mechanical game playing and problem solving. These characteristics are not peculiar to "42"; they are found in many problems of scientific and economic importance. The development of techniques to appropriately cope with these additional complexities should result in the extension of mechanical problem solving to a larger class of problems.

A 2-person board game is a game between two players in which the state of the game is fully displayed by the locations of the board pieces. This type of game possesses two important properties:

1) perfect information: there is no chance element and each player possesses all the information necessary to determine the best line of play.

2) zero-sum: a measure of the desirability of a board situation to one player is necessarily a measure of its undesirability to the opponent.

Checkers, Chess, Kalah, Go-Moku, and Go all fall into this category and have been the major areas of mechanical game playing experimentation.

"42" differs from the two-person board game in several important respects:
1) "42" is a four player, two team game requiring good partnership interaction for quality play.

2) "42" lacks the perfect information property: a chance element exists in that a player sees only his own "hand" and is unaware of the distribution of the other twenty-one dominoes.

3) "42" is not a zero-sum game: the fact that one player may view the situation as good does not necessitate another player's viewing it as bad. This results from imperfect information regarding the dominoe holdings of the other players.

4) "42" bidding requires determination of a bid instead of choice of a move. In the two-person board game, the board situation after each potential move was evaluated to determine the move of maximum advantage. This same procedure does not lend itself readily to "42" bidding since "making a particular bid" does not alter the hand being evaluated. Thus "42" bidding may be defined as a two phase problem, evaluating the hand and choosing the bid as the two distinct phases. Each of these properties of "42" requires special consideration in the design of a "42" bidding machine.

**COGAP-42: The "42" Bidding Machine**

COGAP-42, Computer Game Playing -42, simulates human bid determination in the game of "42". COGAP-42 is a two phase system: 1) the hand is evaluated, 2) the bid is determined. Hand evaluation consists of two general goals,
offensive ability and defensive ability, which are subdivided into appropriate sub-goals. The method of evaluating offensive and defensive goals and their associated sub-goals is via a signature table procedure.

**Signature Table Evaluation (17)**

The first signature table procedure was proposed by Arnold Griffith, an M.I.T. graduate student, and was modified and implemented by A. Samuel (17) in Checker Player II.

A signature table contains a set of $N$ parameters called a signature type; the $N$-tuple of values of the $N$ parameters is a signature. Associated with each signature type is an argument whose value maintains a one-to-one correspondence with the $N$-tuple of parameter values. For a particular evaluation instance, the $N$ measured parameters uniquely determine the signature argument. This argument then designates the entry into a scoring table associated with the signature type; from the table entry, one obtains a score reflecting the desirability or worth of the signature describing the particular instance under consideration. The scores obtained from the various signature tables may then be combined to determine the overall score or evaluation.

A signature table consisting of only one signature type is a one level signature table. Signature types may be combined to form a hierarchial signature table of several levels. In such a multi-level structure, the scores
associated with a signature type are quantized into \( M \) categories; signatures of similar desirability or worth reside in the same category and have the same category number. Each signature type may now itself be considered a single parameter whose value is one of \( M \) category numbers. Subsets of these new parameters are now combined to form higher level signature types whose outputs or scores are once again quantized to obtain parameter values for the next level. This process is continued until reaching a level consisting of only one signature type. This highest level table is not quantized but determines the overall score or evaluation for the particular instance under consideration.

**Sample Signature Table**

A simple two level signature table will illustrate the mechanics of this hierarchial structure. In Figure 2-1, level one of the signature table consists of two signature types. The first scores trump point value and consists of two interacting parameters: TRNUM \((0, 1, \text{ or } 2)\) and TRPPT \((0, 1, 2, \text{ or } 3)\). The second signature type measures trump control value and consists of three interacting parameters: TRNUM \((0, 1, \text{ or } 2)\), TRPCT \((0, 1, \text{ or } 2)\), and TNPTO \((0 \text{ or } 1)\).

The arguments associated with the two signature types must maintain a one-to-one correspondence with the individual signatures; therefore they have a range of twelve and eighteen values respectively. In a particular
instance, the measured signatures might be (2,3) and (2,1,1) and the corresponding arguments 12 and 16. These arguments become the entries into the associated evaluation tables from which one obtains the quantized score for each signature. For the first signature type, the quantized score in this case must be between 0 and 2 and becomes the measured value of the new parameter TRVPT. Similarly the new parameter TRVCT, representing the second signature type, assumes a measured value between 0 and 4. These two parameters now interact and characterize the signature and associated argument of the level two signature type. This argument provides the entry into the associated scoring table from which the final evaluation is obtained. This sample signature table might be considered a measure of overall trump strength. Its final scores could again
be quantized and combined with measures of other factors to obtain an overall evaluation of a hand.

**Basic COGAP-42 Evaluation and Bidding Procedures**

COGAP-42 utilizes two independent signature tables, designated the offensive goal and the defensive goal. The offensive signature table is a four level hierarchial structure. The highest level goal of determining offensive ability consists of three level three sub-goal signature types: trump strength, overall strength, and overall weakness. These in turn each consist of two or more level two sub-goal signature types and so forth until reaching the base level composed of the individual heuristic parameters.

The defensive signature table is also a four level hierarchial structure, the higher level goals similarly consisting of lower level sub-goal signature types.

To determine a bid, COGAP-42 first evaluates its hand. Each suit of length two or more is considered as possible trump by the offensive goal; the resultant hand with this trump configuration is evaluated by the offensive signature table. The offensive goal is satisfied once the trump configuration yielding the greatest offensive score is determined; this suit is designated offensive trump and its offensive score becomes offensive ability. Associated with this is a quantized score designating one of 0^ offensive categories. The hand is then evaluated by the defensive signature table to obtain a measure of
defensive ability and a quantized score indicating one of $D_1$ defensive categories.

Hand evaluation is now complete and the bid determination phase begins. The last previous bid and a designation as to whether by partner or opponent is categorized into one of $L_1$ categories. The offensive, defensive, and last bid category numbers interact to form a signature with an associated argument. This argument serves as the entry into the bidding table from which the appropriate bid is extracted. COGAP-42 then makes this bid, naming "offensive trump" as its chosen trump suit.

Signature Table Features: Non-Linearity and Goal Orientation

The signature table was selected as the evaluation procedure for COGAP-42 because of its non-linearity and its adaptability as a goal-oriented mechanism. The non-linear feature is important because of the many interdependencies of parameters in "42" which cannot be accurately measured by a linear scoring method such as an evaluation polynomial.

The hierarchial structure is preferable to a single level table for two reasons. Human strategy is simulated more closely by a goal directed hierarchial structure than by a single signature type containing all parameters of the system. Secondly, and most important, the extremely large number of signatures within the single signature table would exhaust machine storage facilities. (For the COGAP-42 offensive table alone, the single signature type
would require over $10^{10}$ words of storage.)

The adaptation of the signature table to a goal-directed tree structure effects simulation of human bid determination in "42". The common method of constructing a hierarchial signature table is to develop base level parameters, group these together on the basis of assumed interdependencies to form signature types, and then proceed in the same manner to group together interdependent signature types to form the multi-level structure.

Although its operation remains the same, COGAP-42 views the signature table as an evaluation structure in which general goals are broken down into more specific sub-goals instead of as a structure built up from basic heuristics. This aspect is important and was stressed during construction of the signature tables themselves. The goal-directed process attempts to simulate human bid determination, as it subdivides long range goal evaluations into more specific sub-goals. As each sub-goal is satisfied, the longer range goal evaluations may be attempted until finally the most general goal is satisfied and an overall score determined. Thus COGAP-42 views the hierarchial signature table as a tree of goals and sub-goals. To maintain this distinction and to further simulate human thought, the most general goal or level of the COGAP-42 signature table henceforth will be designated level 0 and subsequent levels will be appropriately re-numbered so that the greatest depth is attained at the heuristic parameter
goals. Thus the COGAP-42 signature tables resemble tree structures with a maximum depth of four. Figures 2-2, 2-3, 2-4, and 2-5 illustrate the goal trees and associated signature tables.

**Imperfect Information and Team Play**

New methods were designed to handle the imperfect information feature of "42". A heuristic was developed to estimate two essential trump probabilities: the probability of losing trump point count and the probability of losing trump tricks. This heuristic is described in Chapter 4. Other probabilities are not calculated outright, but heuristics are employed when distribution of the other dominoes becomes a factor in sub-goal determination.

Good partnership interaction is required for quality play in "42". This is accomplished by COGAP-42 in several ways. The partner's bid or lack of it is itself a parameter of pertinent signature types or sub-goals. Parameters in many cases consider partner's unknown dominoes; this is most evident in the heuristic probability calculations but occurs to a lesser degree in other heuristics. The individual sub-goals at times pertain to the partnership as a whole instead of solely to the individual hand being evaluated. Thus partner is considered throughout COGAP-42 bid determination.
FIGURE 2-2: Offensive Goal Tree

- **Offensive Ability**
  - **Overall Weakness**
  - **Overall Strength**
  - **Trump Strength**

See Figure 2-2a for Sub-goals
See Figure 2-2b for Sub-goals
See Figure 2-2c for Sub-goals
FIGURE 2-2a: Trump Strength Sub-goal Tree
Figure 2-2b: Overall Strength Sub-goal Tree
OVERALL WEAKNESS

PARTNER'S STRENGTH

LEAD POINT COUNT LOSERS

LOSE 5-POINT COUNTERS

LOSE 6-4

LOSE 5-5

CHANCE OF LOSING EACH INDIVIDUAL 5-POINT COUNTER

NON-LEAD POINT COUNT LOSERS

PARTNER'S STRENGTH

NON-HOLD POINT COUNT WEAKNESS

CHANCE OF LOSING NON-HOLD POINT COUNT

UNPROTECTED POINT COUNT WEAKNESS

CHANCE OF LOSING UNPROTECTED POINT COUNT

FIGURE 2-2c: Overall Weakness Sub-Goal Tree
FIGURE 2-3: Offensive Signature Table
FIGURE 2-4: Defensive Goal Tree
FIGURE 2-5: Defensive Signature Table
COGAP-42 and Types of Bidding

Four distinct types of bidding occur in "42":

1) offensive: the player bids because he feels his team can make the contract.

2) defensive: the player feels he is strong enough to set any opposition bid and therefore passes. Alternately, the player feels his team is so weak that it could not possibly set the opponents at this level. He therefore bids hoping to force the opponents to a higher level.

3) informative: the player feels his hand is not of sufficient quality to secure a contract. However, he can provide considerable help to his partner who has not yet bid and therefore makes a bid of 30 to indicate this. In most instances, this situation occurs because the first player does not possess a sufficiently strong trump suit.

4) save-partner: this situation occurs when partner has made an informative 30 bid and the intervening opponent has passed. If the player has a poor hand, his passing may force his partner to play an impossible hand — perhaps one with only two trump. Therefore he may make a bid himself only to choose the lesser of two evils.

It is interesting to note how COGAP-42 reacts to these situations.

Offensive bidding is straightforward: if offensive ability is strong for the present bidding level, the proper
bid is made. Defensive bidding is more complex. If defensive ability is extremely high, COGAP-42 recognizes its chance to set the opponent and, depending upon offensive ability and the bidding level, may choose to pass. However, the other type of defensive bidding --- to force the opponent to a higher level --- has not been achieved in COGAP-42.

COGAP-42 may make an informative bid if partner has not yet bid and it discerns low offensive ability and high help-partner ability. The signature table for determining defensive ability will also measure one's helpfulness if partner obtains the bid. This is evident since the same goals will be effective in either case.

A special procedure is used to achieve proper save-partner bidding. If partner has bid informatively, an intervening bid by opponent has not occurred, and normal bid determination indicates a pass, this save-partner mechanism is activated. This mechanism considers only the best hand and its associated trump suit as determined by the offensive goal. The signature corresponding to the overall trump strength sub-goal is extracted from the offensive goal tree and characterizes the potential trump suit. Its associated argument serves as an entry into a save-partner table from which a bid or pass indication is obtained. Save-partner bidding thus occurs if the trump suit is likely to be better than partner's.

Thus in all but one instance, COGAP-42's bidding procedure parallels human bid determination. Further
evidence of this and detailed analysis of bid simulation is contained in Chapter 5.

Additional Remarks

Several remarks on this game playing mechanism should now be made. The range of values which a basic heuristic parameter assumes or into which a signature type is quantized is a matter of experimentation. Too small a range results in insufficient characterization of parameters and poor playing ability; too large a range severely hampers the learning phase of the project.

It is conceivable that the "42" bidding machine might have been designed as one large signature table with the possible bids as a table parameter. A hand would be evaluated for each possible bid, and the bid with the highest final score selected. This would then have been a one phase system resembling the Samuel Checker Player (17) as opposed to the COGAP-42 two phase system in which the hand is first goal evaluated and then these goals are combined to determine the actual bid.

Several factors lessen the desirability of such a one phase bidding system. Foremost is the fact that such a mechanism would bear little resemblance to human bid determination; the human bidder does not evaluate a hand with respect to each possible bid. Thus the simulation aspects of the project would have been lost. Secondly, such a structure would be far larger than the present signature table; this would occur for technical reasons
concerning the possible bid parameter and its large range of values. This size factor not only would tax machine memory facilities but would also severely hamper if not eliminate successful learning.

**Summary**

COGAP-42, the "42" bidding machine, is a two phase system simulating human bid determination. A particular hand is evaluated by goal-oriented signature table procedures; then the final goal scores are used in combination with other parameters to determine the proper bid. Strategy is introduced by designing the signature tables as evaluation systems in which general goals are broken down into more specific sub-goals instead of as systems built up from basic heuristics. Special heuristics were developed to insure good partnership interaction and to alleviate the difficulties caused by imperfect information.
Chapter 3
"42" AS AN EXPERIMENT IN LEARNING

Why Study "42" as an Experiment in Learning

Research in machine learning has occurred primarily in pattern recognition, theorem proving, and game playing, specifically two person board games. One of the main objectives of the author's research was to investigate a new area of machine learning, with the intention of developing techniques that would facilitate successful learning in a larger class of problems.

"42" bidding was selected as the subject of this research project because it possesses several characteristics common to problems of scientific and economic importance but not previously a part of mechanical learning experimentation:

1) "42" bidding is a two-phase system — evaluating the hand and choosing the bid as the two distinct phases. (The Additional Remarks section of Chapter 2 explains why a one phase system is not desirable for the "42" machine.) Bid determination depends upon hand evaluation but the hand evaluation phase is completely independent of bid determination. Appropriate learning schemes must be coordinated between the two phases.

2) "42" bidding requires special reinforcement techniques — a rating of the goodness of a hand is desired in the hand evaluation phase, not a choice
among alternative moves or bids. Therefore, a comparison of several alternatives does not occur and the common reinforcement technique — to positively reinforce the best choice and to negatively reinforce all others — is not applicable.

3) "42" bidding has imperfect information — this feature requires that more complex heuristics be developed to adequately reflect the probabilities of important dominoe distributions. Otherwise learning will be extremely slow, if at all acceptable.

4) "42" bidding does not provide a store of expert play — the stock of bids determined by non-master players will not always be correct or consistent but from this the machine must learn a good line of play.

The "42" project was undertaken with the intention of developing a coordinated game playing/learning system in which the decision making functions parallel those of humans and in which rote and generalized learning act interdependently to improve the machine's playing ability. A goal-directed bidding procedure was envisioned as a simulation of human bid determination, as a simulation of some aspects of human learning, and as a method of increasing the learning rate by segmenting the overall bidding process into more easily learned sub-blocks. Special techniques are used to handle the aforementioned "42" properties. Layered learning is introduced to increase the rate of learning so that a successful "42" machine may
be developed with a relatively small amount of learning experience.

**Introduction to COGAP-42 Learning**

COGAP-42, the "42" bidding machine, simulates human decision making and improves its playing ability with experience in the bidding of "42". Both generalized and rote learning are utilized in the system, each to its particular advantage. Generalized learning effects improvement of the overall evaluation procedures by analysis of small amounts of data in great detail and abstraction from this data of general notions about what comprises good and bad "42" hands. Rote learning allows very specific information to be accurately gathered for future reference.

**Generalized Learning**

Generalized learning is necessitated in COGAP-42 by the number of possible dominoe distributions and bid sequences that may occur. $10^6$ different hands may be dealt to a player and 733 different bidding sequences may precede his own bid for a total of $8 \times 10^8$ bidding combinations. Within an individual hand, several different hand configurations occur depending upon the choice of trump. Because of the limitations on memory store, search time, and the available stock of non-master play, rote learning alone is insufficient to gather successfully the large amount of knowledge required for good bidding.
The method of effecting generalized learning in COGAP-42 is through the signature table procedure. (The reader is referred to Chapter 2 for a discussion of the construction, use, and advantages of the signature table as an evaluation procedure.)

Checker Player II Signature Table Learning (17)

A.L. Samuel (17) developed the hierarchical signature table as the basic learning scheme of Checker Player II. Associated with each signature are two tallies, A (Approve) and D (Disapprove). Given a board situation, the multi-level signature representations of all possible moves are determined. For each incorrect move, the D tallies of its associated signatures are incremented by one. For the recommended "book" or expert move, the corresponding A signature tallies are incremented by N, the number of non-book moves. Thus positive reinforcement is weighted more heavily than is negative.

Periodically, this accumulated learning is merged into the signature table itself to alter the evaluation procedure. A correlation coefficient, \( C = \frac{(A-D)}{(A+D)} \), is computed for each signature. This coefficient then reflects the goodness of the associated signature as a criterion in move selection. A signature associated primarily with good moves will have a positive correlation coefficient, one associated with poor moves a negative one. The correlation coefficients have a range of values from -1 to +1, scoring the goodness of the associated
signatures as move indicators.

These correlation coefficients are then entered into the signature table as the scores of each associated signature. For the highest level table, these scores become the final board evaluations. In the case of lower level tables of a hierarchial structure, these scores must be quantized to obtain entries into the next table level.

For a given signature type, the signature scores are classified into M categories, where M is the number of values the new signature parameter will assume. Signatures with zero scores or correlation coefficients are placed together in category zero. The positive scores are classified according to rank into \([M/2 - 1]\) positively ranked categories, with an equal number of signatures in each category; the negative scores are similarly classified into the remaining negatively ranked categories. The positive and negative category numbers now serve as values for the new signature parameter and are used to calculate the entry into the next table level.

Learning is accumulated into the A and D tallies for each board situation; however, the overall signature table is altered only periodically. During the early stages of learning, blank entries will occupy much of the table. Information abstracted into the higher level tallies will be determined by the categorization of the lower level tables, but this categorization in many cases will be based

\[1. \left\lfloor K \right\rfloor \text{ means the largest integer } \leq K.\]
upon insufficient learning. These difficulties may be alleviated but not overcome by smoothing of the data tables. Through linear manipulation of the entries and use of the zero-sum checker characteristic, the tables may be smoothed to effect slightly better learning at higher levels and to allow full use of the tables during play despite the blank entries.

This concludes discussion of signature table learning in Checker Player II.

Introduction to COGAP-42 Generalized Learning

COGAP-42 is a two phase system — hand evaluation and bid determination. As described in Chapter 2, signature table procedures are the scoring mechanisms for the goal-oriented hand evaluation phase. These signature tables also act as generalized learning schemes to effect proper interaction and weighting of sub-goals and to achieve a good hand evaluation procedure.

The signature table was chosen as the evaluation and learning procedure for COGAP-42 because it allows non-linear interaction of parameters. In the offensive and defensive goal trees, sub-goal A may be extremely important if sub-goal B has a particular value and of only moderate utility otherwise. This feature could not be adequately reflected by a linear evaluation or learning scheme. Other reasons for selecting a hierarchial signature table scheme are discussed in Chapter 2.
COGAP-42 signature table learning differs significantly from the original signature table learning scheme. Reinforcement, categorization, and smoothing techniques have all been modified for application to "42" bidding.

**COGAP-42 Signature Table Reinforcement: Offensive**

The reinforcement method of Checker Player II (17) must be significantly altered for COGAP-42. The hand evaluation phase requires a rating of the quality of a hand, not a choice among alternative bids or moves. Therefore, an overall comparison of several alternatives does not occur and the common reinforcement technique for generalized learning --- to positively reinforce the best choice and negatively reinforce all others --- is not applicable.

However, during offensive evaluation of a particular hand, several hand configurations --- determined by different trump suits within that hand --- may be analyzed and compared. Within the offensive signature table, the best hand configuration and its associated trump suit may be positively reinforced while all other possible configurations are negatively reinforced. After sufficient learning experience, the offensive evaluation procedure should be capable of analyzing a particular hand and selecting the trump suit yielding the best overall hand configuration.

This reinforcement of the offensive signature table
is insufficient for overall learning of offensive hand
evaluation; it alone will not develop a mechanism capable
of rating a given hand's offensive capability. A comparison
of the quality of different hands must be made, not just a
comparison of different hand configurations within the same
hand. Using the aforementioned reinforcement scheme, a
hand consisting of seven trump would never be compared
with a hand consisting of two trump and five doubles since
these two cases could not occur as hand configurations
within the same hand.

To overcome this difficulty, recourse is made to the
assumed relationship between the initial bid for a hand
and the quality of that hand. If this relationship does
exist and is relatively accurate, a weighted reinforcement
of the best hand configuration may be made via the asso-
ciated bid. In this manner, an informal comparison of the
quality of completely different hands is effected by
comparison of their associated bids and weighted reinforce-
ments. This is the basis of COGAP-42 offensive signature
table reinforcement.

Given a particular hand, the multi-level offensive
signature representations of all possible hand configura-
tions with two or more trump are formed. If a non-pass
bid and trump suit are indicated by the trainer, the
associated signatures of the hand configuration corre-
sponding to the "correct" trump suit are positively
reinforced. This weighted reinforcement is accomplished by
incrementing the A and D signature tallies by ABI and DBI respectively:

**EQUATION 3-1: Offensive General Positive Reinforcement**

**A Bid Incrementation (ABI)**

\[ \text{ABI} = \begin{cases} 
\text{Bid} - 15 & \text{if Bid} \neq 84 \\
30 & \text{otherwise}
\end{cases} \]

**D Bid Incrementation (DBI)**

\[ \text{DBI} = \begin{cases} 
45 - \text{Bid} & \text{if Bid} \neq 84 \\
0 & \text{otherwise}
\end{cases} \]

Thus the greater the bid, the greater will be the Bid Incrementation Average \( \frac{\text{ABI}}{\text{ABI} + \text{DBI}} \). A high (low) Bid Incrementation Average should reflect a hand of high (low) quality. The new tally weighting averages

\[ \frac{A'}{A' + D'} = \frac{(A + \text{ABI})}{(A + \text{ABI}) + (D + \text{DBI})} \]

will contain the positive learning from this new hand.

For each incorrect hand configuration, the associated signatures are negatively reinforced to indicate that these hand configurations are of lesser goodness than indicated by the given bid. This negative reinforcement is accomplished by incrementing the D signature tally by NR as follows:
EQUATION 3-2: Offensive General Negative Reinforcement

Given: \( A = \) signature A tally value
\( D = \) signature D tally value
\( \text{ABI} = A \) Bid Incrementation
\( \text{DBI} = D \) Bid Incrementation

Then:
\[
\text{NR} = \begin{cases} 
1 & \text{if } A=0 \text{ and } D=0 \\
\frac{\text{DBI} \cdot A - (\text{ABI} \cdot D)}{A + \text{ABI}} & \text{if } \frac{\text{ABI}}{\text{ABI} + \text{DBI}} \leq \frac{A}{A+D} \\
0 & \text{otherwise}
\end{cases}
\]

\( \text{NR} \), the negative reinforcement quantity, is dependent upon the signature A and D tally values and therefore must be calculated separately for each signature.

The derivation of the appropriate formula for this negative reinforcement quantity follows from several considerations. If the A and D signature tallies are both zero, this signature has not been encountered previously and \( \text{NR} = 1 \) so that the Tally Weighting Average \( \frac{A}{A + D} \) has a finite zero value. Signature A and D tallies which produce a Tally Weighting Average \( \frac{A}{A + D} \) less than or equal to the Bid Incrementation Average \( \frac{\text{ABI}}{\text{ABI} + \text{DBI}} \) already reflect the lesser value of this signature. These tallies therefore need not be altered and \( \text{NR} \), the negative reinforcement quantity, is 0.

The only complication arises when the Tally Weighting Average is greater than the Bid Incrementation Average; that is, \( \frac{A}{A + D} > \frac{\text{ABI}}{\text{ABI} + \text{DBI}} \) so that a signature
associated with an incorrect configuration is reflected as being of greater desirability than indicated by the bid for the hand. In this case, it is desired that new A and D tallies be formed that contain a reflection of this hand's lesser quality.

To merely increment the A and D tallies by ABI and DBI respectively will of course decrease the Tally Weighting Average $\frac{A}{A + D}$. However, ABI and DBI only indicate the maximum goodness of each signature in this case; it is likely that this incorrect hand configuration is really of far lesser quality than indicated by the ABI and DBI weights. Such a tally incrementation would have the undesirable effect of reinforcing this signature at a higher level than appropriate. This reinforcement, whereby the A tally, as well as the D tally, is incremented, would lessen the rate at which this Tally Weighting Average is affected by future negative reinforcement.

It is therefore desirable to alter the tallies in such a manner that the new Tally Weighting Average $\frac{A'}{A' + D'}$ equals the weighting $\frac{(A + ABI)}{(A + ABI) + (D + DBI)}$ that would result from the aforementioned reinforcement but in such a manner that the A tally remains unchanged. Then the rate of future negative reinforcement will not be decreased. These conditions lead directly to the equation

$$\frac{A}{A + (D + NR)} = \frac{A + ABI}{(A + ABI) + (D + DBI)}$$

Solving for NR, one obtains Equation 3-2, the offensive general negative reinforcement formula.
If a pass bid is indicated, negative reinforcement of the entire hand must occur. For each hand configuration, the associated signature D tallies are incremented by PNR as follows:

**EQUATION 3-3: Offensive Pass Negative Reinforcement**

Given:  
A = signature A tally value  
D = signature D tally value

Then:

\[
PNR = \text{MAX}\left(\frac{19A - 11D}{A + 11}, 8\right)
\]

PNR, the pass negative reinforcement quantity, is dependent upon the present A and D tally values and therefore must be calculated separately for each signature.

This formula for pass reinforcement was determined in an analogous manner to the general negative reinforcement formula. The least allowable offensive positive bid reinforcement results in an ABI of 13 and a DBI of 17. It is necessary that pass reinforcement decrease the Tally Weighting Average \(\frac{A}{A + D}\) by more than would positive reinforcement by the least bid. However, for the same reasons mentioned under general negative reinforcement, the A tallies should not be altered. These considerations lead directly to the equation

\[
\frac{A}{A + (D + PNR)} = \frac{A + 11}{(A + 11) + (D + 19)}
\]

Solving for PNR, one obtains Equation 3-3, the offensive pass negative reinforcement formula. It is desirable to substantially decrease the Tally Weighting Average for
each signature associated with a pass; therefore PNR assumes an experimentally determined minimum value of 8.

It was stated that a hand's initial bid, as determined by the trainer, is utilized in reinforcement. This is not entirely correct. The peculiarities of "42" result in initial bidding gaps; initial bids of 30, 31, or 35 are common but an initial bid between 32 and 34 is an extremely rare occurrence. Also, a good hand might not be bid if it is very strong defensively. For these reasons, the offensive bid given by the trainer is actually the average bid contract the hand is likely to make.

COGAP-42 Signature Table Reinforcement: Defensive

In reinforcing the defensive signature table, only one hand configuration is used since trump is not designated. Therefore no comparisons of even sub-alternatives occur, as was the case within the offensive table. A comparison of the defensive capability of different hands is effected by relating the defensive quality of a hand to the average "42" contract that hand is likely to set.

Given a particular hand for defensive learning, the multi-level defensive signature representation of that hand is formed. The trainer indicates the average "42" bid that hand is likely to set and reinforcement is accomplished by incrementing the A and D tallies of each associated signature by ADI and DDI respectively:
EQUATION 3-4: Defensive Positive Reinforcement

A Defensive Incrementation (ADI)

$$ADI = 42 - \text{(Set Bid)}$$

D Defensive Incrementation (DDI)

$$DDI = \text{(Set Bid)} - 30$$

Thus the lower the bid which the given hand is likely to set, the greater is its defensive ability and therefore the greater is its Defensive Incrementation Average

$$\frac{ADI}{ADI + DDI} \ . \ The \ new \ Tally \ Weighting \ Averages$$

$$\frac{A'}{A' + D'} = \frac{A + ADI}{(A + ADI) + (D + DDI)}$$

will contain the defensive learning from this hand.

Considerable learning accumulates in the offensive and defensive signature tallies. It is possible, even likely, that many a tally will overflow the bits allotted to it. Whenever a tally is to be incremented, an overflow test is made; if positive, both the A and D tallies for that signature are decremented by 20%. This maintains the same Tally Weighting Average $$\frac{A}{A + D}$$ but allows additional learning to be accumulated.

**COGAP-42 Signature Table Categorization**

Periodically, the learning accumulated into the A and D tallies must be used to alter and improve the offensive and defensive signature table evaluation procedures. The Tally Weighting Average $$\frac{A}{A + D}$$ (corresponding to, but differing from, the correlation coefficient $$\frac{A - D}{A + D}$$ of Checker Player II) is computed for
each signature. This Tally Weighting Average assumes a value between 0 and 1 and is entered into the signature table as the score for its associated signature. The greater the Tally Weighting Average, the greater is the desirability of this particular signature.

At the offensive or defensive general goal level (level 0), the Tally Weighting Average becomes the overall evaluation of offensive or defensive ability. At each level of the signature table, these averages must be quantized to obtain entries into the next table level and to form the values assumed by the new signature parameters.

For a given signature type, the signature scores are classified into T categories, where T is the number of values the new signature parameter will assume. This classification is accomplished by a more complex range process than the equi-size category process of Checker Player II (17). This range categorization is designed to lump signatures of like goodness into the same category and yet maintain sufficient discrimination within the signature type for good learning.

Initially, T is the total number of categories for this signature type and the category number N = 0. The maximum MX and minimum MN Tally Weighting Averages within the signature type are determined and from these the categorization range \( R = \frac{MX - MN}{T} \) is calculated. All signatures within the highest and lowest range segments are then classified; signatures with Tally Weighting
Averages between MX and MX-R are assigned to category N and signatures with Tally Weighting Averages between MN and MN+R are assigned to category (T-N-1). The total number of categories T for this signature type is decremented by 2, the category number N is incremented by 1, MX and MN are recalculated for the unclassified signatures, and the range categorization process is repeated until all signatures have been classified.

Range categorization produces more accurate classification of signatures in "42" than the equi-size category process. The Tally Weighting Averages within a signature type do not maintain a relatively uniform distribution of values but instead assume a somewhat step-clustered distribution — the averages occur in clusters along a step function. Each cluster may contain from one to many Tally Weighting Averages. Figures 3-1 and 3-2 illustrate the classifications that would occur from each of the two categorization processes for an actual signature type of COGAP-42. Equi-size categorization places signatures of far different goodness in the same category. Range categorization separates the signatures into goodness ranges; a category is assigned only signatures of approximately similar goodness. Thus better signature discrimination is achieved.

It should be noted that the categorization range decreases as one moves from the outer extremes to the center of the Tally Weighting Average distribution. This varied range was designed into the range categorization
FIGURE 3-1: Equi-size Categorization of Twelve Signatures into Four Categories

Tally
Weighting
Average

Signature Number

FIGURE 3-2: Range Categorization of Twelve Signatures into Four Categories

Tally
Weighting
Average

Signature Number
process to avoid wasting category space on empty gaps in the distribution pattern and to provide better discrimination within the more dense center clusters.

Each signature's category number assigned during the range categorization process is entered into the signature table. These numbers are then used during offensive and defensive hand evaluation and during future learning to form entries into the next level tables.

COGAP-42 Layered Learning and Smoothing Operations

Problems necessitating smoothing of the data tables in the Checker Player II (17) signature procedure also occur in the COGAP-42 learning scheme. However, the introduction of "layered learning" overcomes a majority of these difficulties, increases the speed and accuracy of learning, and eliminates the need for all but minor smoothing of the data tables.

The two problems connected with the COGAP-42 learning scheme are the abstraction of learning at higher levels from insufficient learning at lower levels and the occurrence of blank entries within the data tables. In a hierarchial learning structure, it is important that the initial learned instances be general and basic in nature, rather than unusual cases, since these initial instances will form the "building blocks" for future learning abstracted at higher levels. The rate of learning will be significantly slower if an appropriately graded training sequence is not adhered to. However, such
a training sequence is usually extremely difficult, if not impossible, to accurately construct. This difficulty in higher level abstraction occurs in the COGAP-42 signature table procedure. Although the signature table procedure itself is improved only periodically, learning is constantly being accumulated into the signature A and D tallies. In order to learn a given hand, the multi-level signature representation of this hand is constructed. In each case, the higher level signatures are determined from the category numbers of the signatures at the next lower level. If insufficient learning has occurred, the categorization of the lower level signatures will not have reflected their true desirability and abstraction at the higher levels will result in invalid alteration of the higher level signature tallies. This will occur until sufficient learning has been accumulated to effect relative stability among the lowest level (level 4) signatures and to allow approximate determination of their general goodness and associated category numbers. At this point, relatively valid learning will begin at the next level (level 3) but this learning has the additional, difficult, and time consuming task of eradicating the prior invalid table alterations for this level. Invalid entries will continue to be made at higher levels (levels 2, 1, and 0) until this process of stabilization permeates all but the highest level of the hierarchial structure. This abstrac-
tion problem of hierarchial learning is dependent upon the size of the tables and the number of levels of abstraction; it appears that addition of a table level may almost square the amount of learning required for relative table stabilization.

COGAP-42 time limitations and the relatively small store of training bids required the development of a method for overcoming these abstraction problems. "Layered learning" --- the learning of an entire "layer" of instances at one level before abstraction to the next level --- proved to be a successful technique.

The introduction of layered learning to COGAP-42 required alterations in the learning scheme. In order to learn a given hand, only the lowest level signature representation of each hand configuration is formed. The associated signature tallies are appropriately altered to reflect level 4 learning. The level 4 signature representation together with reinforcement data is condensed into three memory words of the "abstraction store" for future use. This process is repeated for each new hand until a large "layer" of level 4 learning has been accumulated, at which point the abstraction routines assume control. The range categorization routine first categorizes the level 4 signatures; then the level 3 abstraction routine accesses the abstraction store and, for each entry, forms the level 3 signature representation, accumulates level 3 learning in the associated A and D tallies, and
re-stores the condensed level 3 signature representation and reinforcement data. The level 3 abstraction routine maintains control until the entire layer of level 3 learning has been accumulated; level 3 signatures are then categorized and abstraction at the next level begins.

This "layered learning" allows a layer of learning to be accumulated at lower levels before abstraction of this layer to higher levels. Although memory limitations prohibit an extensive layer to achieve complete stabilization of the lower level tables, sufficient learning and generalization for greatly improved higher level abstraction may be achieved with only moderately sized layers.

With the introduction of layered learning, smoothing of the data tables is necessary only to remove blank entries before the start of COGAP-42 bidding. Within each signature type, the sub-goals are ordered according to their experimentally estimated human importance; thus the signatures themselves may be ordered such that the fastest varying parameters are those associated with the less important sub-goals.

Smoothing of the data tables is achieved by substituting for each blank entry the closest non-blank entry within the signature type. Because of the ordering of the signatures, the substituted entry will usually represent a signature differing from the original signature only in the less important sub-goals. This smoothing will not
achieve ideal results, due to large clusters of blank entries and the non-linearity of the parameter interaction. However, with layered learning, it has no affect on the learning process and appreciably affects COGAP-42 bidding only until experience has been accumulated for the large majority of signatures in the table.

This concludes the discussion of generalized learning. In summary, COGAP-42 offensive and defensive goal evaluation was developed through generalized learning. This learning was implemented by signature table procedures, with reinforcement and categorization methods especially designed for "42". Layered learning was introduced to alleviate the hierarchial abstraction difficulties and to eliminate the need for all but minor smoothing of the data tables.

Rote Learning

Rote Learning is the cataloguing in memory of specific instance-response or problem-solution pairs. This type of learning is used in COGAP-42 to accumulate probabilities for use in generalized learning and "42" bidding, to learn the actual bid associated with a general class of hands, and to effect save-partner bidding.

Imperfect information necessitates the heuristic estimation of certain probabilities. The probabilities of losing trump point count and trump control are themselves important sub-goals within offensive goal evaluation.
However the heuristic determination of these probabilities is a time consuming process. Therefore, once these probabilities for a particular trump configuration have been calculated, they are "learned" by the machine and entered into a probability table for future reference.

COGAP-42 is a two-phase system: hand evaluation and bid determination. After a layer of phase one evaluation learning has been accumulated, phase two rote learning of the actual bids begins. A hand is submitted to the machine together with a prior bidding sequence and a correct training bid and associated trump suit. The correct trump hand configuration is evaluated by the offensive goal and the overall hand by the defensive goal to classify the hand into offensive and defensive categories. The triplet — offensive category, defensive category, previous bid category — represents a general class of "42" hands and bidding sequences, and is associated with a bid entry in the final bid table. This associated "bid memory word" is divided into bytes, each byte representing a possible "42" bid. COGAP-42 learns the correct bid for the given hand by incrementing the byte for this triplet. If the correct training bid is a pass, this final bid learning process is automatically repeated by COGAP-42 for each potential trump suit. Thus the best bid for each triplet class will be the one whose associated byte registers the highest count.

Rote learning is also used to develop good save-partner
bidding. (See Chapter 2 for a description of this type of bidding.) Whenever the conditions for possible save-partner bidding are present during phase two learning — the bidding sequence indicates partner has bid informatively and an intervening bid has not occurred — the save-partner learning mechanism is activated in addition to normal bid learning. This mechanism retrieves the overall trump strength signature for the correct offensive hand configuration from the offensive signature table; this signature then represents a very general class of trump suits, including that of the hand being learned. Associated with this signature is an entry in the save-partner table; this entry is divided into two bytes, one indicating a pass and the other a save-partner bid. The appropriate byte is incremented, depending upon whether the trainer-designated bid is pass or non-pass. Thus COGAP-42 learns to relate save-partner bidding to the quality of its trump suit.

COGAP-42 Simulation of Human Learning

COGAP-42 simulates human learning in two major respects. COGAP-42 segmented learning is structured along the same lines as that of the human player, with separate but dependent offensive evaluation, defensive evaluation, and final bid learning blocks. Secondly, the interdependency of rote and generalized learning bears a close resemblance to human learning in "42". Generalized goal
evaluation learning utilizes the calculated and rote learned probabilities; final bid and save-partner rote learning pertain to general classes of "42" hands, as learned and designated by the phase one generalized goal learning. Similarly, the human player can estimate and remember relevant probabilities, can abstract evaluation procedures to general classes of hands, and can learn specific bids associated with these general classes.

**COGAP-42 Hand Generation for Learning**

The hands presented to the machine during the learning phase must closely approximate the occurrence of different hand types during actual "42" bidding. Otherwise the machine will not adjust itself properly to achieve the best decision making apparatus for its environment.

COGAP-42 uses a random hand generator to gather hands for the learning process. For each round of bidding, twenty-eight dominoes must be distributed equally among four players. The dominoes are placed in a vector $V$ of length $N$ where $N$ is initially set to 28. A random number $R$ between 1 and $N$ is generated, domino $V(R)$ is dealt to the first hand, the undistributed dominoes are moved up to the vacated entry, and $N$ is decremented by 1. This process is repeated until the first hand is complete with seven dominoes; then the other hands are each dealt in turn.

The hands accumulated in this manner, together with their correct training bids, are designated the learning
store. This learning store is then used during offensive goal, defensive goal, and final bid learning.

**Summary**

COGAP-42 learning differs significantly from previous game playing machines. These differences include the use of a two-phase learning system, new methods of signature table reinforcement and categorization, and a new learning scheme designated "layered learning".

"42" is a non-board game --- making a particular "bid" does not alter the hand being evaluated whereas in a board game, the board situation after each potential move may be evaluated to determine the move of maximum goodness. This factor resulted in a two phase bidding system and new reinforcement techniques.

The two phase system --- hand evaluation and bid determination --- utilizes both rote and generalized learning. These act interdependently and bear a close resemblance to human learning in "42".

New reinforcement techniques were necessary for the signature evaluation scheme. The hand evaluation phase requires a rating of the quality of a hand, not a choice of alternative bids or moves. Therefore, an overall comparison of several alternatives does not occur and the common reinforcement technique for generalized learning --- to positively reinforce the best choice and negatively reinforce all others --- is not applicable. Recourse was
made to the assumed relationship between the initial bid for a hand and the quality of that hand, and appropriate weighted reinforcement schemes were developed.

A different method of categorization was applied to the COGAP-42 signature tables. The Tally Weighting Averages within a signature do not maintain a relatively uniform distribution but instead assume a somewhat step-clustered distribution. A range categorization scheme was developed to lump signatures of similar goodness into the same category and yet maintain sufficient discrimination within each signature type for good learning.

"Layered learning" was introduced in COGAP-42 to alleviate much of the hierarchial abstraction difficulties of generalized learning. This layered learning increases the speed and accuracy of learning in a hierarchial structure and eliminates the need for all but minor smoothing of the data tables.
Chapter 4
HEURISTICS AND DATA ALLOCATION

Introduction

This chapter consists of two sections: I) heuristic parameters, and II) data allocation. Section I defines N-ary heuristic and N-ary heuristic parameter, discusses the selection of sub-goals and heuristic parameters for use within the COGAP-42 game playing and learning routines, and describes and classifies each individual sub-goal or parameter. The probability heuristic is discussed in detail. Section II describes the representation of data structures and discusses the efficiency and relevancy of each.

SECTION I: HEURISTICS

Heuristic Parameters

"Heuristic" is an ill-defined and controversial term. Popular definitions range from "a rule of thumb limiting the search for a problem solution" to "a rule of thumb resembling human action". A thin line separates algorithms (decision procedures guaranteed to produce a solution if any exists) from heuristics (methods which may or may not provide an acceptable solution) but even this distinction is not always maintained. In order to more precisely characterize the parameters of COGAP-42, the term heuristic is divided into sub-classes and defined inductively.
NOTATION:

P is the problem under consideration.

$S_K$ is a set of sub-heuristics of degree at most $K$

where $S_K$ contains at least one $K$-ary sub-heuristic.

$(P, S_K)$ is the problem $P$ acted upon by a set $S_K$ of
sub-heuristics.

DEFINITION:

A 0-ary sub-heuristic (sub-heuristic of degree 0) is a rule of calculation —- void of assumptions and estimations —- immediately applicable to $P$.

DEFINITION:

An N-ary sub-heuristic (sub-heuristic of degree $N$) is a rule of calculation together with one or more relevant assumptions and/or estimations, all of which are immediately applicable to

1) $P$ if $N = 1$

2) $(P, S_{N-1})$ but not $P$ if $N = 2$

3) $(P, S_{N-1})$ but not $(P, S_{N-2})$ if $N \geq 3$

DEFINITION:

An N-ary heuristic (heuristic of degree $N$) is a calculation applied to $P$ that, when broken down into the maximum string of sub-heuristics, contains at least one sub-heuristic of degree $N$ but no sub-heuristics of degree greater than $N$.

DEFINITION:

An N-ary heuristic parameter (heuristic parameter of degree $N$) is the quantity calculated upon application of an N-ary heuristic to the problem $P$. 
Under this formalized definition of heuristic, much ambiguity is removed. Heuristics are classified into degrees depending upon their complexity. An algorithm falls into the category "heuristic of degree 0"; a true heuristic as used by most researchers in artificial intelligence will be of degree one or more.

Several examples from COGAP-42 will illustrate this classification. The parameter TNUMP measures the number of trump --- it is determined via an algorithm and therefore is a heuristic parameter of degree 0. CENT measures the probability of losing trump. It involves a calculation, with assumptions, applied to the hand being evaluated. CENT is classified as a heuristic parameter of degree 1. PRMW measures the number of primary winners. It is obtained from a calculation applied to the hand, involving estimations, and utilizing the 1-ary parameter CENT. It is therefore a heuristic parameter of degree 2. The sub-heuristics may obviously be layered to any degree of complexity.

The heuristic parameter of degree 0 parallels the parameters of Samuel's several checker playing programs. (16,17) The N-ary heuristic parameter is necessitated by the imperfect information features of "42". Without perfect information, estimations and rules of thumb must be used to calculate complex parameters resembling the considerations of the human player.

The use of the N-ary heuristic in simulation is
easily illustrated. The human player is favorably disposed to the three dominoes 4-4, 4-3, 4-2 in the absence of 6-4, not because of the intrinsic worth of each individual domino but because the entire group contributes as a unit to the capture of the 10-point domino 6-4. Thus a possible N-ary heuristic parameter might be the likelihood of winning the domino 6-4.

The affect of N-ary heuristics upon the rate of learning is readily apparent. Without such heuristics, learning would have to relate to the individual dominoes held, instead of to the affects of these domino holdings. Thus imperfect information would have to be accounted for by the learning itself instead of being considered within the parameter to which the learning relates.

The COGAP-42 structure was determined through analysis of human bidding protocols and questionnaires. In this context, a protocol is a record of a human subject's verbal analysis during a particular task. Such verbatim records afford insight into human thought processes and are therefore useful in devising a simulation system. A sample "42" bidding protocol appears in Figure 4-1. Analysis of these protocols indicated the segmentation of the entire bidding system into two phases, evaluation and bid determination, and the division of the evaluation phase into separate offensive and defensive goals. It was also noted that the human players further subdivided the offensive and defensive goals into sub-goals and that this
subdivision occurred to a depth of several levels. However, the precise nature of the individual lower level sub-goals and the lowest level heuristic parameters was not clearly designated by the protocols although several hypotheses were formed. These were strengthened or rejected on the basis of detailed bidding questionnaires completed by the subjects. A sample questionnaire appears in Figure 4-2.

On the basis of this analysis of human behavior in "42" bidding, the COGAP-42 goal-oriented structure was designed. In the following sections, this breakdown of "42" bidding into component segments and sub-goals is described. Individual heuristic parameters are briefly defined and classified into degrees. Sub-goal descriptions are indented and numbered to indicate the level of each. It should be noted that parameters preceded by (*) are calculated from analysis for the given hand whereas all other goals are determined from the interaction of their own lower level sub-goals. A graphical description of this structure appears in Figures 4-3 and 4-4. Figure 4-5 presents a human subject's analysis and bidding of a "42" hand together with a trace of COGAP-42 analysis and bidding of the same hand.

**COGAP-42 Bidding Structure**

COGAP-42 is divided into two phases: hand evaluation and bid determination. Hand evaluation occurs first in the bidding process, then the phase II bid determination
FIGURE 4-1: Sample Protocol

Hand is 4-3, 1-4-2, 3-0, 4-4, 2-2, 5-4, 1-1

HUMAN SUBJECT:

I have a lot of 4's --- looks like it would be a possibility for trump. I have 4 of them so there's only 3 out but one is 6-4. I have a chance of losing a trump trick if 6-4 doesn't fall and partner doesn't have it. Then I'd possibly even lose another 10 or 5 points. But more likely to fall, I think, so should win all my trump and that means I get 6-4 and 4-1. So should be a good trump suit. Now with that as trump, everything wins but 3-0. Maybe 3-2 goes on 2-2 but likely to lose 3-0 to 3-2. Can't win 5-5 so would probably have to lose 3-0 early before opposition can discard 5-5 on it. I should only lose those 5 points. Looks like a very good hand. Is there any other trump suit? Not really. All have only 2 elements and I know they'll be worse than this real good 4 suit. Don't have any point count and my doubles aren't worth much so I'll have to bid high. Only lose 5 points if 6-4 falls so bid 36.
FIGURE 4-2: Sample Questionnaires

Specific Questionnaire pertaining to Protocol of Figure 4-1.

Q. Is it important to you that you can't win 5-5 outright?
A. It would be better if I could but I can't lose it either unless it is discarded on 3-0.

Q. Would you ever consider a trump suit with only 2 elements?
A. Sometimes have to do so---may not have any other.

Q. Why at the end did you consider your lack of point count?
A. It is necessary to bid higher if I can't set the opposition and since I haven't point count, I won't be able to determine whose tricks the points fall on.

Q. What do you mean by "doubles aren't worth much" at the end?
A. Well, 4-4 --- have too many 4's so opposition will probably trump that; means I wouldn't win 6-4 if playing defense. And 2-2 and 1-1 are very low doubles.

Q. What is the significance of low doubles?
A. Few leads by opponents that would let me win 1-1 or 2-2. I'd have to get the lead some other way and then lead them and I'm not going to get the lead with 4-4.
FIGURE 4-2: (cont.)

General Questionnaire on Trump

1. When do you consider trump? Give as full an explanation as possible.

2. What factors do you consider in determining the value of the trump suit. List as many as possible and then order them in order of importance.

3. How does strength of trump compare in importance to other factors in the hand?

4. How do you decide which trump suit provides the best hand?
FIGURE 4-3: Graphical Description of Hand Evaluation Goals
FIGURE 4-4: Graphical Description of Bid Determination Goals
FIGURE 4-5: A Comparison of Hand Analysis and Bidding by a Human Subject and by COGAP-42.

**HUMAN SUBJECT:** Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Well, let's see what bid this hand might make.

(Subject first considers and discards 5 as a trump suit. Since COGAP-42 considers trump in order, this is moved to the end so as to afford comparison with the machine.)

Have 3 blanks---can consider that as trump. In that case, we will be very unlikely to lose control in trump and almost impossible to lose 5-0 point count. But can only take 5 points in trump. Have only 3 trump---sort of so-so---can use two trump to get opposition trump and 3rd trump to trump in later. Not a bad trump suit.

**COGAP-42:**

Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Enter Phase I Hand Evaluation---Offensive

Select trump = 0 for consideration

Evaluate trump suit

Evaluate trump control worth and point worth

- Probability of losing control once is 19%, twice is 2%
- Probability of losing 5-0 is 15% - 25%
- Scores ability to win trump tricks as 0/4
- Scores ability to win trump point count as 6/9
- Notes 3 trump, and a 10 point non-trump dominoe out that may get sluffed if opposition wins a trump trick

Scores trump control strength as 2/3

Scores trump point strength as 7/9

Scores overall trump strength as 3/5

---

2. A score of A/B means a score of A from 0,...,B where 0 is best and B is poorest score.
FIGURE 4-5: (cont.)

HUMAN SUBJECT: Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

What else have we got. Will win 5-5 and 2-2 definitely --- may walk 5-3 if can trump back in after losing 4-2 --- not too good though since 6-5 and 5-4 are out and likely to be saved. But 5-5 is real good, that's 10 points. And from trump should win 5-0 and maybe 3-2 on 2-2. That only leaves 6-4 and 4-1.

And maybe partner has something.

COGAP-42: Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Evaluate overall strength

Evaluate trick strength

Notes will win 5-5, 2-2, and may walk 5-3 although not very certain, and have an extra trump that can be used for other things than to draw trump

Scores trick strength at 6/8

Evaluate point count strength

Notes will win 5-5

Rates chance of taking 6-4 as 2/2

Notes will surely take 5-0 and also may win 3-2

Rates 5 point winners at 1/4

Scores point count strength at 1/4

Notes partner has not bid

Rates overall strength at 2/4
FIGURE 4-5: (cont.)

HUMAN SUBJECT: Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Well, what do I have to lose. That 4-2 isn't good. Could lose with 6-4 or 4-1 falling, maybe both. Doesn't matter too much if can trump in to walk 5-3 but if lose 6-4 and then opposition leads 3-3, they'll take my 5-3 and maybe 3-2. Then we're down. But that is the worst and not real likely. If 5-3 doesn't walk, points may fall on it.

Hmmm. Well, it isn't a bad hand this way-- if all goes well could make 31 but 30 is safer since then could lose 4-1 and 3-2 or 6-4 and 3-3 with no points on it.

COGAP-42: Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Evaluate Overall Weakness

Evaluate non-lead point count losers

Notes no unprotected point count and only 3-2 as extra loss by opposition leading

Scores non-lead point count weakness at 0/2

Evaluate lead point count losers

Notes that 4-2 loses to 6-4 and 4-1

Scores loss of 6-4 at 3/5

Scores loss of 5 pointers at 1/4

Scores lead point count losers at 3/7

Notes partner has not bid

Evaluates overall offensive weakness at 3/6

Evaluates overall offensive ability for this hand configuration as 9/15
FIGURE 4-5: (cont.)

HUMAN SUBJECT: Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

We'd better look at 2's as trump also. Not real good. Almost certain to lose trump trick and may lose 3-2 then too although it may drop when I lead 2-2. Though maybe I wouldn't lead 2-2 since then when I lose the trump trick, someone might throw a 10 pointer, but not too likely. But should have all trump out with two leads since have 3 of them.

COGAP-42: Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Select trump = 2 for consideration

Evaluate trump suit

Evaluate trump control worth and point worth

Probability of losing control once is 56%, twice is 17%
Probability of losing 3-2 is between 15 and 25%
Scores ability to win trump tricks as 1/4
Scores ability to win trump point count as 6/9
Notes 3 trump, and a 10 point non-trump out that may get slapped if opposition wins a trump trick.
Scores trump control strength as 5/8
Scores trump point strength as 7/9
Scores overall trump strength as 3/5
FIGURE 4-5: (cont.)

HUMAN SUBJECT: Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Now besides trump, 5-5 and 0-0 win and once again maybe 5-3. But have to get rid of that 6-0 before can try and walk it. And again we'll win 5-0, 5-5, and maybe something from partner.

COGAP-42: Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3
Evaluate overall strength
Evaluate trick strength
Notes will win 5-5, 0-0, and may walk 5-3 although not very certain, and have an extra trump that can be used for other things than to draw trump.
Rates trick strength at 6/8
Evaluate point count strength
Notes will win 5-5
Rates chance of taking 6-4 as 2/2
Notes will surely take 5-0 and chance of taking 3-2
Rates 5 point winners at 1/4
Scores point count strength at 1/4
Notes partner has not bid
Rates overall strength at 2/4
FIGURE 4-5: (cont.)

HUMAN SUBJECT: Hand is 0-0,6-0,2-2,4-2,2-0,5-5,5-3

But I still have that off to 6-4. Maybe can throw it away after I lose the trump trick, and only lose 5 points there. But if they lead 6-6 or 4-4, I'll still have to toss it and maybe lose 6-4 along the way. That's too much since also don't know about whether I'll lose points on 5-3. Doesn't look so good this way. Think I prefer it with 0's as trump.

COGAP-42: Hand is 0-0,6-0,2-2,4-2,2-0,5-5,5-3

Evaluate overall weakness

Evaluate non-lead point count losers

Notes no unprotected point count or extra losses by opposition leading

Scores non-lead point count losers at 0/2

Evaluate lead point count losers

Notes that 6-0 loses to 6-4
Rates loss of 6-4 at 3/5
Notes that 5-3 may now lose to 5-0 and trump may lose to 3-2

Scores 5 point loss at 2/4

Scores lead point count losers at 4/7

Notes partner has not bid

Evaluates overall offensive weakness at 5/6

Evaluates overall offensive ability for this hand configuration as 10/15
FIGURE 4-5: (cont.)

**HUMAN SUBJECT:** Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

5's is a possible trump but it's so much worse than any others that we can forget about it.

**COGAP-42:** Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Select trump = 5 for consideration

Evaluate trump suit

Evaluate trump control worth and point worth

- Probability of losing control once is at least 66%
- Probability of losing control twice is 34%
- Probability of losing 5-0 is at least 55%
- Probability of losing 5-5 is 0%

Scores ability to win trump tricks as 3/4

Scores ability to win trump point count as 6/9

Notes 2 trump, and a 10-point non-trump dominoe out that may get sluffed if opposition wins a trump trick

Scores trump control strength as 8/8

Scores trump point strength as 7/9

Scores overall trump strength as 8/8

(COGAP-42 does not immediately discard 5 from trump consideration but continues to evaluate this hand configuration, getting an overall evaluation of offensive ability of 15/15.)
FIGURE 4-5: (cont.)

HUMAN SUBJECT: Hand is 0-0,6-0,2-2,4-2,2-0,5-5,5-3

Now, what do I bid. Chance at 35 but that's really unreal. Too much luck needed, especially with that 4-2 off. If I bid, it has to be 30 or 31.

Well, I've got 5-5 and only one other 5. That would be real good on defense or to help partner if he decides to bid. Also 2-2 and 0-0 but they won't help me gain control. And lots of points out for partner to throw on my tricks. But really all I have is that 5-5, but that's real good, especially with only 5-3.

COGAP-42: Hand is 0-0,6-0,2-2,4-2,2-0,5-5,5-3

Enter Phase I Hand Evaluation —— Defensive
Evaluate overall defensive and informative ability
Evaluate trick strength
Evaluate winning tricks
Notes that have one big double 5-5 and two little doubles, 0-0 and 2-2, and that sufficient 5 leads are out to make 5-5 important.
Rates winning tricks as 4/7
Notes that all point count except 5-5 are out
Notes partner hasn't bid yet
Rates trick strength as 3/5
FIGURE 4-5: (cont.)

**HUMAN SUBJECT:** Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

But that 5-5 is 10 points right there. Not great defense but certainly fairly good and could be of help to partner if get in with 5-5.

**COGAP-42:** Hand is 0-0, 6-0, 2-2, 4-2, 2-0, 5-5, 5-3

Evaluate point count strength

Evaluate win 5-5

Rates 5-5 at 1/3

Evaluate win 6-4

Rates chance of winning 6-4 at 3/3

Evaluate point count held strength

Notes hold 5-5 but no other point count

Rates point count held strength at 3/4

Rates overall point count strength at 4/7

Evaluates overall defensive-help partner ability as 2/4
FIGURE 4-5: (cont.)

**HUMAN SUBJECT:** Hand is 0-0,6-0,2-2,4-2,2-0,5-5,5-3

Well, if I bid 31, then I'll have to walk 5-3 --- but with 30, then maybe can toss it off. And may be able to set 31 and can help partner if he bids 31.

Bid 30 with trump = 0.

**COGAP-42:** Hand is 0-0,6-0,2-2,4-2,2-0,5-5,5-3

Enter Phase II Bid Determination

Notes that first bidder

Selects 0 as trump with offensive evaluation of 9/15

Defensive ability at 2/4

Bids 30 with trump = 0

(Other subjects bid 31 or passed. However, COGAP-42 accounts for this possibility since the pass and 31 bytes of the final bid entry are non-zero. However the byte 30 registers the largest count and 30 is therefore the selected bid.)
uses the results of phase I to obtain the desired bid.

Phase I --- Hand Evaluation

The hand evaluation phase seeks to determine the quality of a given hand with respect to two goals: OFABL (offensive) and DFABL (defensive).

**OFABL (0): OFFENSIVE GOAL**

The offensive goal measures overall offensive ability. The hand configuration, with designated trump suit, achieving the highest offensive score, is designated the offensive hand; its associated score becomes the measure of offensive ability. The offensive goal is determined from the three sub-goals: TRSTR, OVSTR, and OVWKN.

**TRSTR (1)**, an offensive sub-goal, evaluates overall strength of the designated trump suit. It consists of two sub-goals: TRCNT and TRPTS.

**TRCNT (2)**, an overall trump strength sub-goal, measures the overall control strength of the designated trump suit. It consists of three sub-goals: TRPCT, TNUMC, and TENPO.

**TRPCT (3)**, a trump control strength sub-goal measures the value of the trump suit in drawing trump. It consists of only one sub-goal: CENT.

(**)CENT (4)**, a heuristic parameter of degree 1, is a combined measure of the probability of losing trump control once and more than once. The next
section entitled the Probability Heuristic contains a detailed description of this probability calculation. 

(*)TNUMC (3), a trump control strength sub-goal and heuristic parameter of degree 0, measures the number of trump in the designated trump suit.

(*)TENPO (3), a trump control strength sub-goal and heuristic parameter of degree 0, measures the number of 10-point non-trump dominoes that opponents may discard during a loss of control.

TRPTS (2), an overall trump strength sub-goal, measures the overall point count strength of the designated trump suit. It consists of two sub-goals: TNUMP and TRPPT.

(*)TNUMP (3), a trump point count strength sub-goal and heuristic parameter of degree 0, measures the number of trump in the designated trump suit.

TRPPT (3), a trump point count strength sub-goal, measures the value of the trump suit in securing trump point count. It consists of only one sub-goal: PENT.

(*)PENT (4), a heuristic parameter of degree 1, is a combined measure of the probability of losing five, ten, and
fifteen points in trump point count.
The next section entitled the Probability Heuristic contains a detailed description of this probability calculation.

OVSTR (1), an offensive sub-goal, evaluates overall strength of the hand configuration. It consists of three sub-goals: TKTAK, PTTAK, and PBID1.

TKTAK (2), an overall strength sub-goal, measures the overall trick strength of the hand configuration. It consists of three sub-goals: PM, SECY/, and UNBY/.

(*) PMW (3), a trick strength sub-goal and heuristic parameter of degree 2, measures the number of primary winners, or tricks very likely to be taken, by this hand configuration. The calculation of this heuristic parameter entails consideration of many complex factors such as likelihood of being trumped, loss of one dominoe to promote others in the suit, whether high dominoes in a suit are likely to "fall", and so forth.

(*) SECW (3), a trick strength sub-goal and heuristic parameter of degree 3, measures the possibility of taking an extra trick if the dominoes "fall" correctly.
(*)UNBW (3), a trick strength sub-goal and heuristic parameter of degree 3, measures the number of primary winners that are "unbacked" and therefore possibly may be taken by a correct opposition lead.

PTTAK (2), an overall strength sub-goal, measures the overall point count strength of the hand configuration. It consists of three sub-goals: FIVT, SIXT, and FVPT.

(*)FIVT (3), a point count strength sub-goal and heuristic parameter of degree 0, measures the presence or absence of the 10-point dominoe 5-5.

(*)SIXT (3), a point count strength sub-goal and heuristic parameter of degree 1, measures the likelihood of winning the 10-point dominoe 6-4.

(*)FVPT (3), a point count strength sub-goal and heuristic parameter of degree 1, measures the likelihood of winning the several 5-point dominoes.

(*)PBID1 (2), an overall strength sub-goal and heuristic parameter of degree 0, measures whether partner has bid.

OVWKN (1), an offensive sub-goal, evaluates overall weakness of the hand configuration. It consists of three sub-goals: NLPLS, LPTLS, and PBID3.
NLPLS (2), an overall weakness sub-goal, measures non-lead point count loss weakness. It consists of three sub-goals: UNPPT, PNTKO, and PBID1.

UNPPT (3), a non-lead point count loss sub-goal, measures unprotected point count weakness. It consists of only one sub-goal: UNPRO.

(*)UNPRO (4), an unprotected point count sub-goal and heuristic parameter of degree 2, measures likelihood of losing held point count as a result of opposition leads. The point at which control may be lost is an important consideration in this calculation.

PNTKO (3), a non-lead point count loss sub-goal, measures non-held point count weakness. It consists of only one sub-goal: PNOTK.

(*)PNOTK (4), a non-held point count loss sub-goal and heuristic parameter of degree 2, measures the likelihood of losing non-held point count on opposition leads. The point at which control may be lost is an important consideration in this calculation.
(*)PBID1 (3), a non-lead point count loss sub-goal and heuristic parameter of degree 0, measures whether partner has bid.

LPTLS (2), an overall weakness sub-goal, measures lead point count loss weakness. It consists of three sub-goals: LS55, SXFLS, and FIVLS.

(*)LS55 (3), a lead point count loss sub-goal, and heuristic parameter of degree 0, measures the possibility of losing the 10-point domino 5-5.

SXFLS (3), a lead point count loss sub-goal, measures the possibility of losing the 10-point domino 6-4. It consists of only one sub-goal: LS64.

(*)LS64 (4), a lose 6-4 sub-goal and heuristic parameter of degree 1, designates the type of dominoes held relevant to a loss of 6-4.

FIVLS (3), a lead point count loss sub-goal, measures 5-point domino loss and weakness. It consists of only one sub-goal: LS5.

(*)LS5 (4), a lose 5-point dominoes sub-goal and heuristic parameter of degree 1, measures the likelihood of losing each individual 5-point domino.
(**)PBID (2), an overall weakness sub-goal and heuristic parameter of degree 0, measures the value of partner's bid.

DFABL (0): DEFENSIVE GOAL

The defensive goal measures overall defensive ability of the hand. It consists of two sub-goals: DTRST and DPTST.

DTRST (1), an overall defensive sub-goal, measures defensive trick strength. It consists of three sub-goals: TETAK, NUMPT, and PBID.

TETAK (2), a defensive trick strength sub-goal, measures the possibility of winning defensive tricks. It consists of four sub-goals: DFGE4, DFLDS, and DFSEC and DFLS4.

(**)DFGE4 (3), a win defensive trick sub-goal and heuristic parameter of degree 0, measures the number of doubles held in suits 4, 5, and 6.

(**)DFLS4 (3), a win defensive trick sub-goal and heuristic parameter of degree 0, measures the number of doubles held in suits less than 4.

(**)DFLDS (3), a win defensive trick sub-goal and heuristic parameter of degree 0, measures the number of doubles held in suits 4, 5, and 6 in which the opponents can have at most two leads.
(*)DFSEC (3) , a win defensive trick sub-goal and heuristic parameter of degree 0, measures the number of suits in which the hand has second round control only.

(*)NUMPT (2) , a defensive trick strength sub-goal and heuristic parameter of degree 0, measures the number of point dominoes available for partner to discard on winning tricks.

(*)PBID (2) , a defensive trick strength sub-goal and heuristic parameter of degree 0, evaluates partner's bid.

DPTST (1) , an overall defensive strength sub-goal, measures point count defensive strength. It consists of three sub-goals: DF55, DTK64, and DPNTS.

(*)DF55 (2) , a point count defensive strength sub-goal and heuristic parameter of degree 0, measures the possibility of winning the 10-point domino 5-5.

DTK64 (2) , a point count defensive strength sub-goal, measures the possibility of winning the 10-point domino 6-4. It consists of only one sub-goal: DF64.

(*)DF64 (3) , a win 6-4 sub-goal and heuristic parameter of degree 0, designates the dominoes held relevant to defensive winning of 6-4.
DPNTS (2), a point count defensive strength sub-goal, evaluates the point count dominoes held. It consists of four sub-goals: FPRPT, SPRPT, UPRPT, and PBID.

FPRPT (3), a point count dominoe sub-goal, measures fully protected point dominoe strength. It consists of only one sub-goal: DFPRO.

(*)DFPRO (4), a fully protected point count dominoe strength sub-goal and heuristic parameter of degree 0, designates the type of fully protected point count held.

SPRPT (3), a point count dominoe sub-goal, measures the singly protected point dominoe strength. It consists of only one sub-goal: DFSPR.

(*)DFSPR (4), a singly protected point dominoe strength sub-goal and heuristic parameter of degree 0, designates the type of singly protected point count held.

UPRPT (3), a point count dominoe sub-goal, measures the unprotected point count dominoe strength. It consists of only one sub-goal: DFUPR.
(*)DFUPR (4), an unprotected point dominoe strength sub-goal and heuristic parameter of degree 0, designates the type of unprotected point count held.

(*)PBID (3), a point count dominoe sub-goal and heuristic parameter of degree 0, measures partner's bid.

Phase II --- Bid Determination

The bid determination phase uses the results of Phase I hand evaluation and its own heuristic parameters to determine the appropriate bid for the given hand and prior bidding sequence. The interacting parameters in bid determination are OFABL (offensive goal), DFABL (defensive goal), and PRVBD. Only the last parameter remains to be described.

(*)PRVBD, a heuristic parameter of degree 0, describes the last non-pass bid of the prior bidding sequence, characterizing it as to quantity and as to partner or opponent.

Probability Heuristic

The calculation of the offensive heuristic parameters CENT and PENT merits detailed description. These heuristics measure the probability of losing trump control once and more than once and the probability of losing five, ten, and fifteen points in trump point count.

These probabilities are extremely difficult to deter-
mine accurately. Both the overall distribution of trump dominoes and the distribution of individual trump point counters must be considered. Imperfect information and the resultant playing strategy by the several players substantially affects the number of tricks and the number of points won by a team. For instance, consider the trump holdings illustrated in Figure 4-6. North is leading. Several lines of play are possible with very different outcomes. If North leads 6-6, the obvious playing sequence by the four players is 6-6, 6-2, 6-5, 6-0, dropping North's partner's 6-5. Now East will be able to win his 10-point counter 6-4 and the final result will be that N-S wins 2 points, E-W wins 11. However if North were to lead 6-1 instead, two possible playing sequences may occur: 6-1, 6-4, 6-5, 6-0 or 6-1, 6-2, 6-5, 6-0. North would then play 6-6 on the second round of trump. In either case, N-S would win the 10-point counter 6-4 on either the first or second round and the final result would be far different: N-S wins 13 points, E-W wins 0 points. However since North cannot see the other players' trump holdings,
he cannot know that this latter line of play will be successful. Thus each individual player's strategy has a definite and substantial affect upon the winning of trump points.

The present COGAP-42 method of calculating these several trump probabilities for a given hand has been quite successful. In the following description of this heuristic, Hand I is the dominoe hand for which the probabilities are being determined. Hands I and III comprise Team A; Hands II and IV comprise Team B.

In the calculation of the CENT and PENT probabilities for a given hand (Hand I), several important heuristic assumptions are made:

1) Trump will be led as long as possible. Player I leads.
2) Perfect information regarding the other hands' trump holdings is afforded each player.
3) Discards will always be worthless dominoes.
4) Each team will attempt within this framework to maximize the number of trump points won.

Assumptions (1), (3), and (4) limit the probability determination to consideration of trump only. During actual play, it might be advisable not to lead trump and instead trump opposition tricks or, for a variety of reasons, to accept the possible loss of trump points in exchange for other advantages. Assumption (2) allows the determination of the best playing strategy for each hand. These
heuristic assumptions are extremely important; their affect upon the several probabilities will be discussed later.

To determine CENT and PENT for the given Hand I, five counters are initially set to zero; they are FIVE, TEN, FIFT, CONT1, and CONT2 referring to a loss of 5, 10, or 15 points in trump point counters and a loss of control at least once and more than once, respectively. For a given distribution of the remaining trump dominoes among the other three players, a look-ahead evaluation is used. A tree of possible moves by each player is expanded until one team can no longer play trump. At this point, the overall point loss to team A (both for trump point count and for tricks) is scored for each line of play. Mini-maxing (see chapter 1 for a description of a mini-max procedure) determines the correct line of play in which Team A moves to minimize this point loss and Team B to maximize it. This line of play is the best playing strategy for the given trump distribution. Counters are then incremented to reflect this:

1) FIVE, TEN, and FIFT are incremented by one if this playing strategy involves the loss of a 5-point dominoe, a 10-point dominoe, and a total of 15 points respectively.

2) CONT1 and CONT2 are incremented by one if this playing strategy involves the loss of control at least once and more than once respectively.

This process is repeated for each possible distribution of
the remaining trump dominoes among Hands II, III, and IV. The individual counters updated during this process are then divided by the total number of such trump distributions to determine the CENT and PENT probabilities for Hand I.

Several remarks regarding this probability determination should be made. In order to limit memory requirements, the entire move tree is not expanded at once. Instead a preorder move expansion-erasure with minimax procedure is used. Preorder tree expansion follows from Knuth's (13) definition of preorder tree traversal.

The preorder move expansion-erasure with minimax procedure is defined recursively:

1) List the root (first move) of the first tree.
2) Apply preorder move expansion-erasure with minimax to the subtrees of the first tree.
3) Apply preorder move expansion-erasure with minimax to the remaining trees.
4) If a root of the listed tree has no subtrees, score the line of play indicated by the move tree expansion.
5) If all subtrees of a root have been listed at some past time, erase the root and apply minimax to determine the correct score for the previous node.

An example will clarify this procedure. Figure 4-7 shows a full move tree expansion with minimax. Under the preorder expansion-erasure with minimax procedure, the same result is obtained using less memory space:
FIGURE 4-7: Three Ply Look-Ahead With Minimax

Final Score = 6
at Node A
FIGURE 4-7(a-i): Three Ply Look-Ahead: Preorder Expansion
Erasure with Minimax

Figure 4-7a

Figure 4-7b

Figure 4-7c

Figure 4-7d

Figure 4-7e

Figure 4-7f

Figure 4-7g

Figure 4-7h

Figure 4-7i
a) Root A (the first move) is listed. Preorder expansion-erasuere with minimax is now applied to the first subtree of root A. The move tree is expanded (Figure 4-7a). Root E has no subtrees and this line of play is scored as 1.
b) All subtrees of root E have been expanded (there are none). Root E is erased and minimax yields a score of 1 for root D (Figure 4-7b).
c) The move tree is expanded (Figure 4-7c). Node F has no subtrees and this line of play is scored as 0.
d) All subtrees of root F have been expanded. Root F is erased and minimax retains a score of 1 for root D (Figure 4-7d).
e) All subtrees of root D have been expanded. Root D is erased and minimax yields a score of 1 for root B. Similarly root B is erased and root A receives a score of 1 (Figure 4-7e).
f) Preorder expansion-erasuere with minimax is now similarly applied to the second subtree of root A (Figures 4-7f, 4-7g, 4-7h, 4-7i). The final score for the correct line of play is 6 --- the same as in the full move tree expansion case.

It should be noted that in no case were more moves listed at one time during preorder expansion-erasuere with minimax than the number of levels in the entire tree structure. The tree depth pre-determines the maximum number of moves to be retained in memory at any one time.
This procedure therefore requires only a fraction of the memory space necessary for a full move expansion. Preorder expansion-erasure with minimax is most effective as a space saver when the tree being expanded is very bushy.

A team version of the alpha-beta algorithm is applied during the preorder expansion-erasure with minimax procedure. This algorithm indicates lines of play, often entire subtrees, that will have no affect on determination of the correct line of play and therefore need not be expanded and investigated. The alpha-beta algorithm is well documented in the literature (17) and will not be discussed in this chapter; it is briefly summarized in Appendix II. The team version of the alpha-beta algorithm merely substitutes two teams for the usual two players. The rest of the algorithm is identical to the two player case.

The heuristic assumptions made during the probability calculations have minimal affect upon the usefulness of the CENT and PENT quantities. With the exception of perfect information, the heuristic assumptions provide a framework for calculating these specific probabilities while removing extraneous consequences considered by other heuristic parameters. The perfect information assumption, on the other hand, does affect the calculations. However, with each player afforded the same advantage, the discrepancies between point and control loss counters associated with the perfect information determined lines
of play and the point and control loss counters resulting from an imperfect information human strategy are small over the total number of trump distributions considered. The alterations that do occur in the final probabilities are small and ordinarily will not materially affect the category to which the trump hand is assigned.

Two possibilities for future experimentation in this area are being considered. The first would retain imperfect information during the probability calculations, devising heuristic strategies for choosing a player's move. The difficulty here is that such heuristic strategies would in many cases be a function of the probabilities being calculated. Secondly, the construction of a similarity heuristic, to reduce the number of remaining trump distributions considered, would be very desirable. It is anticipated that this heuristic would categorize the many possible distributions into similarity classes and that point and control loss calculated for one member of the class would be representative of all members of that class.

**Summary**

The sub-goals and heuristic parameters for COGAP-42 were structured and selected to simulate human bid determination. An inductive definition of heuristic and heuristic parameter was presented and used to classify the COGAP-42 heuristic parameters according to their
degree of complexity. The calculation of probabilistic heuristic parameters was achieved through a look-ahead play evaluation with a perfect information strategy. Two suggestions were made for future experimentation with such probabilistic heuristics.

SECTION II: DATA ALLOCATION

COGAP-42 game playing and learning requires the maintenance and easy accessibility of large amounts of data. Data structures were selected both for their efficiency in memory utilization and for their applicability to the purposes of the data itself.

An efficient and useful representation of the domino hand under consideration is essential since it is referenced many times during calculation of each heuristic parameter. The data structure finally selected was particularly well-suited to calculation of "42" heuristic parameters; the importance of this hand representation merits a full discussion of its advantages and disadvantages.

A 7x2 array named CHAND was formed to represent the given hand. Each suit was assigned two words of the array. Suits 1 thru 6 are represented by \( CHAND_{A,1} \) and \( CHAND_{A,2} \) where \( A \) is the particular suit; since zero subscripts are not allowed in the Rice Computer Language, suit 0 was given words \( CHAND_{7,1} \) and \( CHAND_{7,2} \). The first word associated with a suit --- the suit holding word---
FIGURE 4-8: CHAND Representation of Hand with no trump suit

Hand is 4-4, 5-4, 4-2, 3-3, 3-2, 2-1, 1-0

FIGURE 4-9: CHAND Representation of Hand with Trump Suit = 4
is a bit representation (bits 0-7) of the elements held in that suit; the second word — the suit lead word — is a bit representation (bits 0-7) of all possible leads by any player in that suit. The double of a suit is indicated by setting bit 7; all other elements are indicated by setting the corresponding bit. Thus the ordering of the elements of a suit is maintained in the CHAND words, with the top dominoes of a suit at the left and the lowest at the right.

An example will clarify this hand representation. Figure 4-8 shows the CHAND representation of the dominoe hand 4-4, 5-4, 4-2, 3-3, 3-2, 2-1, 1-0, without a trump suit declared. Dominoe 3-3 is an element of suit 3; it is a double and is therefore indicated by a 1 in bit 7 of word CHAND_3, 1. Dominoe 1-0 is an element of suits 0 and 1; it is indicated by a 1 in bit 1 of word CHAND_7, 1 and in bit 0 of word CHAND_1, 1. The other elements are similarly represented. The second column of the CHAND array represents all dominoe leads; 2-0, 2-1, and 2-2 are all leads of suit 2 — therefore bits 0, 1, and 7 of CHAND_2, 2 are all set to 1, even though only 2-1 is an element of the hand being represented.

Figure 4-9 presents the CHAND representation of the same hand with suit 4 declared as trump. Several changes result from the designation of trump; these changes will be understood only by those familiar with "42". Since suit 4 is trump, dominoes 5-4 and 4-2 are trump dominoes only; they are no longer elements of suits 5 and 2
respectively — therefore bit 4 of CHAND\textsubscript{5,1} and bit 4 of CHAND\textsubscript{2,1} are both set to 0. Alterations also occur in lead indicators. Al elements of suit 4 are trump dominoes; therefore all are possible leads and CHAND\textsubscript{4,2} indicates this. Dominoes 5-4 and 6-4 are no longer elements of suits 5 and 6 respectively — therefore they cannot be leads in those suits and bit 4 in CHAND\textsubscript{5,2} and CHAND\textsubscript{6,2} is set to 0.

This data structure for hand representation has proven to be a significant help in the special hand manipulations required for "42" analysis and in the determination of "42" heuristic parameters. The advantages derived from it are many:

1) Winners and high dominoes within a suit may be easily determined by shifting and bit tests. This is because the CHAND data structure maintains "42" ordering of the dominoes; moving from bit 7 to bit 0 of a suit word, one moves from the top dominoe or double to ever lower dominoes — until reaching bit 0, representing the lowest element of a suit.

2) A void in a suit is easily evidenced by the all zero CHAND suit holding word. This prevents many useless calculations.

3) The presence or absence of specific point counters is immediately determined by testing the proper bit of the appropriate suit holding word.

4) A count of the number of dominoes held in a suit is
obtained in one instruction through the BCT machine
language operation.

5) Extraction of several pieces of data at a time is
easily facilitated by the machine language logic
operations.

6) Declaration of trump is easily performed: all resultant
alterations in suit holdings and possible suit leads
are evidenced in the CHAND data structure by
appropriate bit changes.

7) The presence of the suit lead words permits the hand's
leads, the hand's non-leads, and the possible
opposition leads in a suit each to be determined with
one logical instruction.

The only difficulty encountered with the CHAND data
structure is the representation of suit 0 in word 7,
necessitated by the absence of zero subscripts. However,
this is minor compared with the many advantages gained
from this method of hand representation.

The probability store is designed for efficient
storage and easy accessibility of the learned trump
suit probabilities. Associated with a given trump holding
is the packed word TRPST_A,B of the probability table
TRPST, where A is obtained from the suit number and B is
a compacted version of the CHAND suit holding bit
configuration. This removes the unused bit in each suit
word --- for example, bit 4 in suit 4 may be removed since
4-4 is a double and therefore indicated in bit 7.
Suits 2 and 3 occupy the same set of words within the table since examination of their ordering and point count shows they are identical as trump suits. With this structure, the probability store is easily accessed for a particular trump suit without additional storage requirements.

The remaining data structures within COGAP-42 have few unusual characteristics. Entry into signature tables generally is obtained directly from the calculated signatures. A few of these tables are compacted if a large number of impossible combinations would otherwise occupy storage facilities --- in this case, the entry must be obtained through a short search procedure.

This concludes the discussion of the COGAP-42 data structures. All were designed to efficiently utilize memory and provide easy access of the stored data. The representation of the hand is most important and was chosen to facilitate the calculation of "42" heuristic parameters and "42" data manipulations.
Chapter 5

RESULTS

Introduction

This chapter evaluates the results of COGAP-42 experiments in game playing and learning and presents suggestions for future extension of these experiments. Four areas of COGAP-42 evaluation are considered:
1) improvement and convergence, 2) affects of layered learning, 3) simulation of human bid determination, and 4) accuracy of probability calculations and range categorization. Each is discussed, together with actual results of the bidding machine, and critically analyzed. The last section of this chapter then discusses proposed improvements to COGAP-42 and possible extensions for future experimentation.

Improvement and Convergence

Although COGAP-42's learning experience did not include a sufficient number of hands to obtain full convergence, significant improvement of the machine's playing ability was achieved. This improvement was evaluated by several different methods.

The first evaluation run consisted of 400 hands or 100 rounds of bidding. For each hand in a round, the machine was presented with the correct previous bidding sequence and asked to bid and name trump. To do this, it was anticipated that the machine would consider the previous
bids, partner's bid, and the quality of its own hand.

A COGAP-42 bid is scored as the best possible if one of the following conditions is satisfied:

1) The bid is Pass and is deemed correct by trainer.
2) Both the bid and declared trump are deemed correct by trainer.

Table 5-1 scores the machine bidding via BBPP: Best Bid with Pass Penalty.

$$BBPP \text{ score } = \frac{B + P \cdot I}{H}$$

where $B = $ number of "best possible" machine bids
$H = $ total number of hands bid
$I = $ number of incorrectly passed hands
$P = $ penalty number

Since approximately 65% of the hands result in Pass bids --- by virtue of the hands and "42" rules --- 65% bidding accuracy could be obtained by merely passing every hand. To remove this factor, a penalty $P$ of -2 is added to the number of correct bids for each improperly passed hand.

Table 5-1 also scores the machine bidding on two segments of this evaluation run:

1) Hands that should be passed
2) Hands that should be bid

Segment 1 is scored via BBPP with the penalties obtained from the overall evaluation run. Segment 2 is scored via BBPP with no penalties ($P = 0$).

These results definitely show considerable improvement of the machine's bidding ability as its learning
experience increases. COGAP-42 quickly learns to determine Pass hands correctly. Its bidding of non-pass hands improves with learning but is not completely mastered over the maximum 800 hands presented for learning. Future convergence of this segment is discussed later in this section.

All of this supports the conclusion that the learning mechanism of COGAP-42 does result in a learning machine that significantly improves its ability as its learning experience increases.

In order to further analyze convergence of COGAP-42 learning, it is desirable to consider a sample containing as large a percentage of non-pass hands with as wide a range of bids as possible. A division of this sample into meaningful segments may then be made and each analyzed regarding convergence.

Such a sample is obtained by listing each of the 400 hands as first bidder of a round. An increase in non-pass hands and a much wider range of bids is noted. This sample removes from consideration factors outside the immediate hand. However, it nonetheless is acceptable for approximating convergence possibilities because:

1) Bid determination for the initial hand of a round is ordinarily more difficult than for later hands since a wider range of choices exists.
2) Improved initial bidding implies improved hand evaluation which in turn implies improved overall bidding.
3) Improvement in overall bidding has thus far paralleled improvement in initial bidding and there appears no reason to suspect a drastic change in this.

Table 5-2 reports results of the initial bidding evaluation run scored via BBPP. It also reports results on six segments of this evaluation run:

1) Hands that should be passed.
2) Hands that should not be passed --- scored on a correct selection of trump.
3) Hands which should not be passed.
4) Hands for which the correct bid is 30 or 31.
5) Hands for which the correct bid is ≥ 32.

Segment 1 is scored via BBPP with the penalties obtained from the overall evaluation run. Segment 2 is scored merely as a percentage of correct trump declarations. Segment 3 is scored via BBPP with no penalties (P = 0).

In segments 4-5, an increase in the number of bids within a specified range should not in itself be allowed to suggest improved accuracy within this range. Therefore a penalty P of 1/2 and 1/3 for segments 4 and 5 respectively is assessed for each incorrect bid within the segment range. This penalty is somewhat arbitrary but was based on the chance of successful guessing.

The four data points for each segment are connected by straight lines and charted by solid lines in Figure 5-3. Analysis of this chart reveals that COGAP-42 quickly mastered the determination of which hands to pass and
convergence of this type of learning was easily obtained. Determination of the correct trump suit for non-pass bids is also mastered. 30 and 31 bid hands improve to a high degree of accuracy and can be assumed to improve relatively little henceforth. 32-36 bids are less accurately determined and 84 bids almost never properly made.

The segments which improve and converge most rapidly are those which comprise the largest percentage of the machine's learning experience. Pass bids and the determination of trump occur most frequently during machine learning; 30 and 31 bids occur relatively often, 32-36 bids occur far less frequently, and bids higher than 37 occur only about 1% of the time.

This order of improvement and convergence of the various segments leads to the hypothesis that additional learning would improve bid determination in the latter segment. This would cause a small improvement in initial bid determination and, paralleling this, a small improvement in overall bid determination.

This theory is further supported by two other analyses. The first involves tracing the table entries used during bid determination of hands in the latter segment. It is found that at least one, and in most cases several, of the signature entries has little or no learning accumulated in its A and D tallies. The category numbers are therefore based upon insufficient data and do not accurately reflect the desirability of the given
signatures. It is further found that the few incorrect bids that occur in the other segments are in many cases also a result of this factor.

The second analysis involves an initial bid evaluation run for a sample consisting of the hands learned by the machine. The results are charted in Figure 5-3 by broken lines and compared with the results for the previous sample. Analysis of these charts reveals that as learning is accumulated, the curves for the two samples approach one another. In the segments for which convergence is obtained, or nearly reached, the two sample curves essentially meet.

As the amount of learning is increased, the affects of the first learned hands are generalized and diluted by further learning. Accuracy in bidding these learned hands slowly approaches the accuracy in bidding hands not included in the learned instances. Convergence implies that additional learning will not alter bidding accuracy; therefore once convergence is obtained, the bidding accuracy in these two samples will be approximately equal.

In the 32 or higher bid segment, the curves for the two samples are still quite far apart. This is the result of few learning instances being related to this segment. As additional learning is undertaken, the two curves must move together. Dilution of the affects of the original learned instances is related to greater generalization of learning. Therefore, it is reasonable to predict that not only will the affects of the learned instances be diluted
causing the learned sample curve to fall but also better generalization will be obtained causing the unlearned sample curve to rise. However, since this segment constitutes a small percentage of any sample run, the improvement in overall bidding accuracy will probably be only about 4%.

Thus it may be concluded that COGAP-42 has achieved convergence in its bidding of the large majority of hands but that its bidding of 32 or higher bid hands will still improve with learning, causing a small improvement in overall bidding accuracy. The author did not continue learning for more than 800 hands because of the time required for trainer analysis of additional hands and the number of such hands necessary for further significant improvement.

Another method of viewing the game playing ability of COGAP-42 is to rate each bid as to its degree of goodness. This will necessarily be a subjective determination, but it provides an interesting characterization of "42" bids. Five classes were designated:

Class 1: Best Bid for the hand.
Class 2: A good bid for the hand
Class 3: Not a good bid, but not a bad bid either
Class 4: A bad bid for the hand
Class 5: A completely and totally unreasonable bid for the hand

Classes 1 and 2 would contain bids by good "42" players;
Class 3, bids that might be made by amateurs; and Classes 4 and 5, bids by those unfamiliar with the game.

Figure 5-4 graphs the results for a sample of 400 hands consisting of 100 rounds of play. Only rarely does COGAP-42 make a Class 4 or 5 bid and only 9% of the time a bid of less than good quality. These graphs are a further indication of the success of this machine in learning to bid "42" hands.

Layered Learning

The ability of layered learning to significantly increase the speed of learning by alleviating abstraction difficulties in a hierarchical structure is demonstrated by the degree of COGAP-42 success achieved with only 300 learned instances, presented in two layers of 400 hands each.

To further illustrate the significance of this layered learning, an additional learning run was made. 200 hands were presented to the machine for learning. However, instead of presenting these hands all in one "layer" or group, the 200 learning instances were divided into 10 layers of 20 hands each. After all 200 hands had been learned, an evaluation run for a sample of 400 hands was made. Table 5-5 contrasts the resultant bidding accuracy with that obtained after a learning run of 200 hands presented in a single layer.

This experiment shows that a significant decrease in learning results from use of the smaller learning layers,
even though the total number of learned instances remains the same. This is caused by hierarchical abstraction difficulties. In all four categories of evaluation, the learning run for 200 hands divided into 10 layers yields less accuracy than the learning run for 200 hands in a single layer, and in three of these evaluation categories, the difference is quite large.

Time did not permit further experimentation in this area. It seems natural to assume however that the use of one-hand layers (200 hands presented in 200 layers of 1 hand each) would even further decrease the learning rate as a result of greater difficulty in hierarchical abstraction. Carrying this still further, the complete elimination of layers (i.e., move the learning for each hand up the entire hierarchical structure immediately but re-categorize the tables only at intervals) can only be expected to cause a far greater deterioration in learning.

Thus it is concluded that the use of learning layers is an extremely useful method of increasing the rate of learning in a hierarchical learning structure, and is responsible to a large degree for the success of COGAP-42.

Simulation of Human Bid Determination

COGAP-42 was conceived in part as a goal-oriented game playing program simulating human decision making functions. This two phase, goal-oriented system resembles human bid determination in "42".
Figure 4-5 in Chapter 4 presents a human subject's analysis and bidding of a "42" hand together with a trace of COGAP-42 analysis and bidding of the same hand. This particular hand was selected because it offers consideration of two reasonable trump suits and has a potential for more than one acceptable bid. It should be noted that the COGAP-42 analysis is presented in order of its goal structure; for reasons of efficiency and use of similar learning subroutines, this order does not precisely resemble the order of computation.

Analysis of Figure 4-5 in Chapter 4 reveals that COGAP-42 simulates to a large degree the decision making of the human player. The COGAP-42 goal structure definitely parallels human bid determination in "42". Factors rated as good or poor by the human player are correspondingly given high or low sub-goal scores by COGAP-42. Lower level goals appear to interact to determine higher level goals in much the same manner as the human player combines sub-factors to determine an overall evaluation of the hand.

However, variations do occur. The human subject interweaves goals considered separately by COGAP-42 or dismisses certain goals as irrelevant to the particular hand. The major difference occurs when COGAP-42 continues to fully consider a hand configuration, with an extremely poor sub-goal (usually trump), that is discarded immediately by the human subject. Also the order of consideration of
trump suits is not identical. It would be desirable for future versions of COGAP-42 to overcome these discrepancies.

The Probability Heuristic and Range Categorization

The ability of the probability heuristic to approximate the several "42" point and control probabilities and the effectiveness of range categorization in classifying the signature tables is reflected in the success of the COGAP-42 bidding machine.

Analysis of the learned probability tables indicates that the probability heuristic provides an extremely accurate approximation of the desired probabilities; minor variations occur but only in very rare cases do they result in an improper categorization of a probability.

Analysis of the various signature tables indicates that range categorization is ordinarily successful in placing signatures of similar goodness in the same category. Occasionally, a signature rated near a range break point will be poorly categorized, but this does not occur often enough to affect the overall success of range categorization.

Thus the probability heuristic and range categorization successfully perform their assigned functions.

Summary

COGAP-42 experimentation has resulted in a learning machine that significantly improves its "42" bidding ability as its learning increases. The high degree of accuracy
obtained suggests that convergence is close at hand; further analyses support the conclusion that this convergence has been obtained for the segment types into which the large majority of hands fall. However these analyses suggest future improvement in one segment comprising about 6% of any random sample. This should result in a 3-4% overall improvement in bid determination.

The success of this learning machine indicates the success of the special reinforcement techniques developed for use in "42" learning; these techniques should be easily extendable to other non-board situations. Layered learning is shown to be an effective method for increasing the learning rate in a hierarchical learning structure. The probability heuristic successfully approximates the probabilities necessitated by imperfect information. Range categorization results in appropriate classification of the signature tables.

As a game-playing machine, COGAP-42 is quite successful in choosing the best bid; rarely does it make a bid that would be considered "very bad". COGAP-42 simulates human hand evaluation and bid determination in most respects; however, this simulation is not as close as desired and offers definite room for improvement.

**Future Research**

Several improvements and extensions of this game playing and learning experimentation suggest themselves. The probability heuristic is successful at approximating
the desired point and control loss probabilities; however, this heuristic is very time consuming. The construction of a similarity heuristic, to reduce the number of remaining trump distributions considered, would be very desirable. It is anticipated that this heuristic would categorize the many possible distributions into similarity classes and that point and control loss calculated for one member of a class would be representative of all members of that class.

Improvement of COGAP-42 simulation of human bid determination should be attempted. This requires the construction of heuristics to discard consideration of irrelevant goals, to consider trump suits in order of probable quality, and to dismiss from consideration hand configurations in which a sub-goal is so poor as to force the overall hand evaluation to be much worse than previously considered configurations.

COGAP-42 learning offers several interesting possibilities for interactive learning of two systems. A heuristic ‘42’ playing machine might be developed. This machine would then be used to play out hands, the results then being used to reinforce the bidding machine. In this way, the bidding machine should eventually reflect the ability of the playing machine. The ultimate goal would be to develop two learning machines, bidder and player, which would be forced to interact with one another in order to improve themselves. However, this will be difficult to accomplish and constitutes a project of major scope and challenge.
TABLE 5-1: COGAP-42 Bidding: Scores

Sample = 400 Hands comprising 100 rounds of "42"

<table>
<thead>
<tr>
<th>Number of Hands Learned</th>
<th>50</th>
<th>200</th>
<th>400</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Bidding</td>
<td>.39</td>
<td>.66</td>
<td>.67</td>
<td>.78</td>
</tr>
<tr>
<td>Segment 1:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hands That Should Be Passed</td>
<td>.46</td>
<td>.76</td>
<td>.76</td>
<td>.86</td>
</tr>
<tr>
<td>Segment 2:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hands That Should Be Bid</td>
<td>.24</td>
<td>.40</td>
<td>.50</td>
<td>.60</td>
</tr>
</tbody>
</table>

TABLE 5-2: COGAP-42 Bidding: Scores

Sample = 400 Hands for initial bidding

<table>
<thead>
<tr>
<th>Number of Hands Learned</th>
<th>50</th>
<th>200</th>
<th>400</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Bidding</td>
<td>.27</td>
<td>.56</td>
<td>.65</td>
<td>.78</td>
</tr>
<tr>
<td>Segment 1:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hands That Should Be Passed</td>
<td>.31</td>
<td>.70</td>
<td>.74</td>
<td>.88</td>
</tr>
<tr>
<td>Segment 2:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trump Selection</td>
<td>.37</td>
<td>.74</td>
<td>.83</td>
<td>.94</td>
</tr>
<tr>
<td>Segment 3:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hands That Should Be Bid</td>
<td>.17</td>
<td>.27</td>
<td>.43</td>
<td>.58</td>
</tr>
<tr>
<td>Segment 4:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hands For Which Correct Bid is 30 or 31</td>
<td>.05</td>
<td>.05</td>
<td>.31</td>
<td>.60</td>
</tr>
<tr>
<td>Segment 5:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hands For Which Correct Bid is ♠ 32</td>
<td>-.32</td>
<td>-.16</td>
<td>.04</td>
<td>.15</td>
</tr>
</tbody>
</table>
FIGURE 5-3a: COGAP-42 Overall Bidding
- Sample = 400 Learned Hands for Initial Bidding
- Sample = 400 New Hands for Initial Bidding

FIGURE 5-3b: COGAP-42 Bidding of Hands That Should Be Passed
- Sample = 400 Learned Hands for Initial Bidding
- Sample = 400 New Hands for Initial Bidding
FIGURE 5-3c: COGAP-42 Trump Selection

- Sample = 400 Learned Hands for Initial Bidding
- Sample = 400 New Hands for Initial Bidding

Score vs. Number of Hands Learned

FIGURE 5-3d: COGAP-42 Bidding of Hands That Should be Bid

- Sample = 400 Learned Hands for Initial Bidding
- Sample = 400 New Hands for Initial Bidding

Score vs. Number of Hands Learned
FIGURE 5-3e: COGAP-42 Bidding of Hands For Which Correct Bid is 30 or 31

Sample = 400 Learned Hands for Initial Bidding
Sample = 400 New Hands for Initial Bidding

SCORE

Number of Hands Learned
FIGURE 5-3f: COGAP-42 Bidding of Hands For Which Correct Bid is ≥ 32

Sample = 400 Learned Hands for Initial Bidding
Sample = 400 New Hands for Initial Bidding

SCORE

Number of Hands Learned
FIGURE 5-4a: Classification of COGAP-42 Bids by Degree of Goodness: OVERALL BIDDING

Sample = 400 Hands: 100 Rounds of "42"

Class 1 is best
Class 5 is lowest

FIGURE 5-4b: Classification of COGAP-42 Bids by Degree of Goodness: Hands That Should Be Bid

Sample = 400 Hands: 100 Rounds of "42"

Class 1 is best
Class 5 is lowest
TABLE 5-5: COGAP-42 Bidding: Scores
Evaluation for Layered Learning

Sample = 400 Hands for initial bidding

<table>
<thead>
<tr>
<th></th>
<th>Layer 1</th>
<th>Layer 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Hands Learned</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Number of Layers</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Number of Hands per Layer</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>Overall Bidding</td>
<td>.56</td>
<td>.34</td>
</tr>
<tr>
<td>Segment 1:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hands That Should Be Passed</td>
<td>.70</td>
<td>.41</td>
</tr>
<tr>
<td>Segment 2:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trump Selection</td>
<td>.74</td>
<td>.72</td>
</tr>
<tr>
<td>Segment 3:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hands That Should Be Bid</td>
<td>.27</td>
<td>.19</td>
</tr>
</tbody>
</table>
CONCLUSIONS

This work has developed a game playing/learning machine for bidding in "42", a dominoes game differing in many respects from previous areas of mechanical game playing experimentation. The major differences are: 1) non-board, 2) non zero-sum, 3) imperfect information, and 4) partnership interaction. These factors required that the machine be designed as a two phase system --- hand evaluation and bid determination --- with rote and generalized learning acting interdependently. This design proved successful, as evidenced by the resultant bidding accuracy. Thus mechanical game playing and learning have been extended to an entirely new type of game.

A result from this experimentation in the area of game playing is the introduction of strategy by using the signature table as a tree of goals and sub-goals. COGAP-42 views the signature table as an evaluation system in which general goals are broken down into more specific sub-goals instead of as a system built up from basic heuristics. In this way, a goal-oriented apparatus is developed and structured along the same lines as human decision making. Ch. 2

Several results in the area of learning machines were obtained. New reinforcement techniques were developed for the signature evaluation scheme. The hand evaluation phase requires a rating of the quality of a hand, not a choice of alternative bids or moves. Therefore, an overall
comparison of several alternatives does not occur and the common reinforcement technique for generalized learning —- to positively reinforce the best choice and negatively reinforce all others --- is not applicable. Recourse was made to the assumed relationship between the initial bid for a hand and the quality of that hand, and appropriate weighted reinforcement schemes were developed. This new reinforcement of the signature table may well be extended to other non-board games and, more significantly, to learning in other fields. Ch. 3.

A new method of signature table categorization was developed. The Tally Weighting Averages within a signature type do not maintain a relatively uniform distribution but instead assume a somewhat step-clustered distribution. A range categorization scheme was developed to lump signatures of similar goodness into the same category and yet maintain sufficient discrimination within the signature type for good learning. This should provide better categorization in other problems having non-uniform distributions within signature types. Ch. 3.

A third result in the area of learning theory was the development of layered learning to alleviate the hierarchial abstraction difficulties of generalized learning. This layered learning increases the speed and accuracy of learning in a hierarchial structure and eliminates the need for all but minor smoothing of the data tables. Ch. 3.
Several results were obtained in the design of heuristic parameters. The terms heuristic and heuristic parameter were inductively defined and classified according to degree of complexity. This should remove much of the ambiguity surrounding use of these terms and affords a method of comparison of different levels of heuristics. A probability heuristic was designed to calculate several probabilities necessitated by imperfect information. This imperfect information was also a major feature of several other heuristic parameters. Ch. 4
Appendix I
RULES AND PLAY OF "42"

Introduction

The following terms and definitions and the general structure of the rules are taken from the game of "bridge" (6) and adapted for use in "42". Preliminary definition of terms will facilitate presentation of "42" rules and sample play.

Definitions

BIDDING: The process of determining the "42" contract.

BID: a declaration of intent to win at least a specified number of points during "42" play.

CALL: any bid or pass.

CONTRACT: the attempt by declarer's side to win at least the number of points specified in the final bid, with the specified suit as trump.

DEAL: the process of distributing the twenty-eight dominoes evenly among the four players.

DECLARER: the player making the last non-pass call.

DEFENDER: a member of the side opposing declarer.

DOUBLE: a dominoe whose two halves are of identical face value.

FACE VALUE: the sum total of spots on a half dominoe.

FOLLOW SUIT: to play a dominoe of the same suit that has been led.

HALF: one of the two halves into which each dominoe is divided.
HAND: the set of seven dominoes originally dealt to a player or any portion remaining during "42" play.

LEAD: the first dominoe played on a trick.

OPPONENT: a member of the other side.

PARTNER: the other member of one's own side or partnership.

PASS: a call indicating that a player chooses not to bid.

PLAY: to remove a dominoe from one's hand and contribute it to a trick.

ROTATION: the clockwise order in which each player receives a turn to call, deal, or play on a trick.

SIDE: any player and his partner.

SUIT: any one of the seven face values of a dominoe half, ranging from 0 to 6.

TRICK: a unit consisting of one dominoe from each player.

TRUMP: a member of the suit specified by declarer in the contract.

TURN: the opportunity for a player to make a call or play.

The Game of "42" --- Rules

The Players

"42" is played by four players with a set of twenty-eight dominoes. The back design of each dominoe is identical.

The Dominoes

Each dominoe consists of two halves containing from zero to six clearly discernible spots; the total number of spots on a half dominoe is the face value of that half. The set of twenty-eight dominoes contains one and only one
dominoe for each possible combination of halves.

**Suits**

With the following exceptions, each dominoe is a member of two distinct suits --- one for each of its halves.

1) A double is a member of only one suit, since its two halves are identical.

2) A dominoe that is a member of the trump suit is not a member of any other suit.

3) A dominoe lead is a member only of the suit represented by its dominoe half of highest face value.

**Rank of Dominoes**

The dominoes of each suit rank as follows:

1) The double is the highest dominoe of the suit.

2) The remaining members of a suit rank in descending order of the face value of the other dominoe half.

**Teams**

The four players divide into two teams. Each player occupies a position opposite his partner, so that the rotation alternates between members of opposing teams.

**The Dealer**

The set of dominoes is placed face down on the table. Each player draws a dominoe and the player drawing the dominoe with the highest sum of the face values of the two dominoe halves is designated first dealer. If two players tie for the highest sum total, the one with the highest face value of a dominoe half becomes first dealer. The right to deal passes in rotation.
The Deal

The dealer mixes the dominoes as they lie face down on the table. The other three players then each draw seven dominoes, the dealer taking the remaining seven dominoes.

The Points

42 points may be won during play of the game. These are divided into trick points and counter points.

Trick points: one point is awarded for each of the tricks won by a side.

Counter points: ten points are awarded to a side for each 10-point dominoe that is a member of a trick won by that side. The 10-point dominoes are 5-5 and 6-4. Five points are awarded to a side for each 5-point dominoe that is a member of a trick won by that side. The 5-point dominoes are 5-0, 4-1, and 3-2.

The total of these points is 42.

Bids

The rules regarding correct bids are as follows:

1) A bid must name a number of points between 30 and 42 inclusive or some multiple of 42.

2) A bid must be greater than the last previous bid, if any.

3) If the last previous bid is less than 42, the maximum allowable bid is 84; otherwise the maximum allowable bid is an increment of 84 over the last previous bid.

4) The last player in rotation is required to bid if no other player has done so.
The Bidding

Bidding begins after the dealer has obtained the seven dominoes of his hand. The dealer makes the first call and each player in rotation makes one and only one call. Bidding is complete when each player has made a call.

The Contract

The contract is the last non-pass call of the bidding sequence and declarer is the player making that call. The contract is successfully made if declarer's side wins the number of points designated by the contract, or 42 points, whichever is less.

Beginning of Play

Declarer names the trump suit and makes the first lead.

Sequence of Play to a Trick

The player who leads may play any dominoe from his hand. Each player in rotation must then contribute a dominoe to the trick. Each player must follow suit if possible; otherwise he may play any dominoe from his hand.

The trick is won by the player playing the highest ranking trump dominoe or, if no trump are members of this trick, the highest ranking dominoe of the suit led. The player winning the trick leads to the next trick.

Tricks

A member of the side winning a trick gathers the dominoes comprising that trick and places them beside him face up in a row separate from any other trick. However, if the contract is 42 or higher, the dominoes are stacked
such that only members of the last two tricks may be examined.

**Completion of Play**

Play ends when declarer's side has succeeded in making the contract or is no longer capable of doing so. The latter occurs if the other side has won \((42 - C + 1)\) points, where \(C\) is the number of points declarer's side must win.

**Marks**

The unit of score is a mark. A contract less than 42 carries a score of one mark, a contract of 42\(N\) a score of \(N\) marks. The score associated with a contract is earned by declarer's side if the contract is successfully made; otherwise, it is awarded to the other side.

**Game**

A game is won by the first side to accumulate seven marks.

**Special Types of Bids**

A bid of 30 is often used as an informative bid to one's partner. The player feels his hand is not of sufficient quality to successfully make a contract. However, he can provide considerable help to his partner who has not yet bid and therefore makes a bid of 30 to indicate this. In most instances this situation occurs because the player does not possess a sufficiently strong trump suit.

A save-partner bid of 31 may be made when partner has bid informatively and the intervening opponent has passed.
If the player has a poor hand, his passing may force his partner to play an impossible hand, perhaps one with only two trump. Therefore he may make a 31 bid only to choose the lesser of two evils.

The Game of "42" --- Sample Play

The following hand is intended as an illustration of the rules and play of "42". It is very simple. Player A is dealer. The holdings of each player are shown in Figure A.

<table>
<thead>
<tr>
<th>A</th>
<th>5-5</th>
<th>5-1</th>
<th>4-4</th>
<th>6-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>2-2</td>
<td>6-2</td>
<td>5-3</td>
<td>5-4</td>
</tr>
<tr>
<td>C</td>
<td>0-0</td>
<td>6-0</td>
<td>2-0</td>
<td>5-2</td>
</tr>
<tr>
<td></td>
<td>4-2</td>
<td>4-3</td>
<td>3-1</td>
<td></td>
</tr>
</tbody>
</table>

FIGURE A: A deal of "42" hands to four players.

Bidding
Player A: 30  Informative bid
Player B: Pass
Player C: 31  Save-partner bid
Player D: Pass
Player C is the declarer and names suit 0 as trump.

<table>
<thead>
<tr>
<th>Play</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player C: 0-0</td>
<td>Draw Trump</td>
</tr>
<tr>
<td>Player D: 3-0</td>
<td>Follow suit --- do not contribute a 5-point dominoe (5-0) to opponent.</td>
</tr>
<tr>
<td>Player A: 1-0</td>
<td>Follow suit</td>
</tr>
<tr>
<td>Player B: 4-0</td>
<td>Follow suit</td>
</tr>
</tbody>
</table>

**Player C wins trick worth one point.**

<table>
<thead>
<tr>
<th>Play</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player C: 6-0</td>
<td>Draw remaining trump</td>
</tr>
<tr>
<td>Player D: 5-0</td>
<td>Follow suit---forced to give opponent a 5-point counter</td>
</tr>
<tr>
<td>Player A: 4-4</td>
<td>Can no longer follow suit. Tells partner to lead suit 4.</td>
</tr>
<tr>
<td>Player B: 5-3</td>
<td>Can no longer follow suit---plays worthless dominoe.</td>
</tr>
</tbody>
</table>

**Player C wins trick worth 6 points**

<table>
<thead>
<tr>
<th>Play</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player C: 4-2</td>
<td>Leads suit suggested by partner.</td>
</tr>
<tr>
<td>Player D: 4-1</td>
<td>Must follow suit.</td>
</tr>
<tr>
<td>Player A: 6-4</td>
<td>Plays highest ranking dominoe in suit 4 since 4-4 was discarded.</td>
</tr>
<tr>
<td>Player B: 5-4</td>
<td>Follows suit.</td>
</tr>
</tbody>
</table>

**Player A wins trick worth 16 points.**

<table>
<thead>
<tr>
<th>Play</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player A: 5-5</td>
<td>Leads a 10-point dominoe</td>
</tr>
<tr>
<td>Player B: 6-2</td>
<td>Can no longer follow suit.</td>
</tr>
<tr>
<td>Player C: 5-2</td>
<td>Follows suit.</td>
</tr>
<tr>
<td>Player D: 6-5</td>
<td>Follows suit.</td>
</tr>
</tbody>
</table>

**Player A wins trick worth 11 points.**

Play ends since declarer's side has made the 31 contract.
Appendix II

ALPHA-BETA ALGORITHM

The alpha-beta algorithm (17) is a tree pruning procedure for eliminating investigation of branches of the tree that analysis indicates will have no affect on the final outcome of minimaxing.

Two values, alpha and beta, are maintained. A board score must be greater than the alpha value if that board position is to be considered by the player making the analysis; but it must also be less than the beta value if the opponent is to allow consideration of that board position. The idea behind this is simple. If the board score does not exceed the alpha value, then the player has a move leading to a more desirable board position of score alpha. If the score is not smaller than the beta value, the opponent has a more desirable previous move (from his point of view) allowing the player a choice of final board position scores of at most beta.

Alpha values are associated with board positions from which the player moves, beta values with board positions from which the opponent moves. Initially, the alpha values are all set to $-\infty$, the beta values to $+\infty$. As a board position is evaluated, its score is compared with the alpha(beta) value associated with the board position two levels up the tree. If the score is not greater(less) than this value, this entire branch of the tree may be discarded. If not discarded, the score is then compared with the
beta(alpha) value one level up the tree. If the score is less(greater) than this value, the board score now becomes the new beta(alpha) value associated with this position. As one moves down new branches of the tree, the alpha or beta values from two levels up are moved down as the new alpha or beta values associated with these board positions. This process is continued throughout the tree. The final alpha value associated with the initial board position will be the final score.

The precise implementation of alpha-beta tree pruning depends upon the evaluation and minimax procedures. As an illustration of this algorithm, consider an evaluation procedure in which boards are scored as integer values. The greater the score, the better is the board for the player about to move. The smaller the score, the better is the position for his opponent. Figure B-1 depicts a simple three ply look-ahead procedure with terminal board scores. Initially, all alpha values are set to -\infty, beta values to +\infty. Board position d is evaluated as +9. This score is compared with the beta value at b and found to be smaller. It is therefore compared with the alpha value at c, found to be larger, and +9 now becomes the new alpha value at c. Board position e is evaluated as +4. This score is compared with the beta value at b and found to be smaller. It is therefore compared with the alpha value at c, found to be smaller, and no changes are made. Board f is similarly compared with no resultant changes. All final board positions from c have been evaluated. Figure B-2
depicts the present situation.

The alpha value at c now becomes the score for board position c, is compared with the alpha value at a, and found to be larger. It is therefore compared with the beta value at b, found to be smaller, and the new beta value at b is now +9. Figure B-3 depicts the present situation.

Moving to investigate the next branch from board b, the alpha value at g assumes the alpha value from two levels up, which is still $-\infty$. Board position h is evaluated as +13. This score is compared with the beta value at board b, found not to be smaller, and therefore this entire branch of the tree is discarded. The beta value of +9 at b now becomes the score for board position b. There is no beta value two levels up from board b so this comparison is null. The score for board b is compared with the alpha value at a, found to be larger, and the new alpha value at a becomes +9. The present situation is depicted in Figure B-4.

This procedure is repeated for the other branch leading from board a, until a final alpha score at a is obtained.

The alpha-beta algorithm is particularly useful in very bushy trees. Researchers have devoted considerable time to determining the most efficient order for consideration of the various branches to effect maximum pruning benefits.
FIGURE B-1: Three Ply Look-Ahead Procedure

FIGURE B-2: Three Ply Look-Ahead Procedure
FIGURE B-3: Three Ply Look-Ahead Procedure

```
ALPHA
-\infty

a

b

BETA
+9

BETA
+\infty

k

c

ALPHA
+9

ALPHA
-\infty

g

d

e

f

h

i

j

9
4
9
13
2
-3

m

n

-2

-1

p

q

4
3
```

FIGURE B-4: Three Ply Look-Ahead Procedure

```
ALPHA
+9

a

b

BETA
+9

BETA
+\infty

k

c

ALPHA
+9

ALPHA
-\infty

g

ALPHA
-\infty

el

ALPHA
-\infty

o

d

e

f

h

i

j

m

n

-2

-1

p

q

4
3
```
Bibliography


