# 10

# An Experiment on Inductive Learning in Chess End Games

R. S. Michalski and Pericles Negri

Department of Computer Science
University of Illinois at Urbana-Champaign

## INTRODUCTION

Further progress in the application of computers to many practical fields seems to depend heavily on the success in implementing learning and inductive processes within machines. For example, to develop a consultation system for medical or plant disease diagnosis, prognosis and decision making in general, it is very desirable, perhaps even necessary, to be able to 'teach' the system through examples of correct and/or incorrect decisions, rather than by precisely describing the decision process in its full generality and then transforming this description into a computer program.

A similar situation exists in computer chess. The development of computer programs playing at the master level (especially the end games) seems to be a formidable task if the programs are not eventually able to learn and improve on their decision making rules through the specific examples of games, rather than by being explicitly told all the rules.

Due to easy access to human knowledge about chess and the relative simplicity of testing the results, chess is one of the most attractive testing domains for inductive inference programs. This report presents first results from an experiment on the application of an inductive learning program called AQVAL/1 developed at the University of Illinois, to chess end games. The experiments were to infer from examples the classification rules which distinguish the win position from the draw position in the single pawn end game (i.e., white king and pawn against black king).

The problem of representing the human knowledge about this end game has been studied by Tan (Tan, 1972). The major and exceedingly significant result of his work is a comprehensive program which solves all known cases of this game (including all examples from Awerbach (Awerbach, 1958) and Fine (Fine, 1941).

His approach, however, has certain strong limitations which make it not very prospective for the more complex end games. As Tan (*loc. cit.*) admits himself:

> "The program does not have any capability of learning, modifying the advice or discovering useful predicates and action schemes. At the moment it is unable to explain its own moves or interpret the opponent's moves ('what is the threat?' 'what does he want?'). It also does not recognize similarity of positions to avoid repetition of analysis."

There has not yet been much done on the implementation of learning processes within the chess playing programs. An interesting effort in this direction has been reported recently by Pitrat (Pitrat, 1974).

In order to clarify the current goals of this work, it is useful to distinguish between different levels of learning, depending on what has to be given to a program learning decision making and what the program is able to do itself:

0-level: (Memorizing facts and rules)

The program has to know:

a) The representation space for the problem (Michalski, 1973), i.e., the program has to know which descriptors* to use in characterizing objects (e.g., situations in chess), and how to measure them,
b) a procedure (a decision rule) for computing the decision class (e.g., win or draw, good or bad move, etc. in chess) to be assigned to an object, based on the description of this object.

The program is able:

c) to memorize and execute the procedure to compute the decision class for any object based on its description.

(A learning process here involves simply the memorization of the methods of measuring descriptors and the decision algorithm.)

1-level: (Learning a decision rule from examples)

The program has to know:

a) the representation space (as in 0-level),
b) examples of objects of different decision classes and the decision classes they belong to.

The program is able:

c) to determine a generalized decision rule for computing a decision for any object,
d) to memorize and evaluate the decision rule for any object.

---

*A *descriptor* is a (unary or n-ary n=2,3,4, . . .) function from the objects or their parts into a set of 'descriptor values'. Unary descriptors are often called features.

2-level: (Learning descriptors and decision rules from examples)

The program has to know:

a) a partial representation space (i.e., to know some basic descriptors of the objects, not necessarily relevant to the specific problem, but which contain sufficient information for creating a new relevant representation space),
b) examples of objects and the correct decision classes they belong to.

The program is able:

c) to create a new problem-oriented representation space (e.g., 'discover' new descriptors) which is more adequate for the given family of decision classes,
d) to infer the generalized decision rules involving new descriptors,
e) to memorize and evaluate the inferred rule for computing decisions for any object.

One can distinguish further levels of learning but will stop here since they are not relevant to this paper. According to the above classification, Tan's work belongs to level 0. The work presented here belongs to level 1.

## AQVAL PROGRAMS

AQVAL/1 programs constitute a package of programs for inductive inference and machine learning.

The following programs are presently in operation:

AQ7 — which infers an optimized description of one decision class in relation to other classes, based on given event sets. The program permits the user to define different optimization functionals, various modes of program operation and some other parameters.
(Michalski, 1973; Larson and Michalski, 1975)

AQ8 (Uniclass) — which determines an optimized description of each decision class separately, under the constraint that the 'degree of generalization' of the description will not exceed a certain value
(described in an internal report)

AQ9 — which optimizes a given set of $DVL_1$ formulas* according to a certain optimality functional
(Cuneo, 1975)

*$DVL_1$ formula stands for a disjunctive simple variable-valued logic formula (Michalski, 1975).

SYM-1 which determines symmetry (with regard to a set of variables) in variable-valued functions and creates $DVL_1$ formulas with symmetric selectors
(Jensen, 1975)

In the experiments reported here, we used only AQVAL/1–AQ7 program. The program is well described and documented (Michalski, 1973; Larson and Michalski, 1975) and therefore we will not discuss it here.

## DESCRIPTION OF THE EXPERIMENTS

### Relation to Tan's work

The end game is a part of the chess game for which a good deal of knowledge is available, and it is often possible to tell whether a move is correct or not. Like humans, machines should use that knowledge to play end games in contrast to the exclusive use of search. An important application of this idea is the program written by (Tan, 1972) to play the single-pawn end game (white king and white pawn against black king). In this program, knowledge is organized in a binary decision tree which associates values of predicates of the board positions with certain decisions. The non-terminal nodes of the tree correspond to predicates of the positions and the terminal nodes (leaves) correspond to ordered pairs <VALUE,ACTION>. VALUE indicates if the position is WINNING (whites win), DRAWING or UNDEFINED. ACTION determines the action scheme that leads to a correct move for that position. To play the game in a given board position the program goes through the decision tree until a terminal node is reached. In other words, the position is classified according to its predicates and the action scheme associated with its class is retrieved. If the VALUE of the terminal node is UNDEFINED, then the game tree is searched depth-first) until the first position that returns a value (winning for the whites and drawing for blacks) is reached. It is important to notice that the knowledge in this program is represented not only by the predicates of the positions but also by the order in which these predicates appear in the decision tree.

Due to the lack of learning capabilities and other drawbacks, Tan's approach is difficult to apply to more complicated end games. The organization of knowledge in a decision tree is by no means a simple task. It requires a great deal of knowledge of chess and even for a chess specialist it would be time consuming because there is no systematic way to do it. As Tan affirms, the organization of the decision tree for the single-pawn end-game was obtained through a trial and error process. It seems to us that the learning process defined by AQVAL could overcome this problem. The fact that the formulas generated by AQVAL programs can be made optimal or quasi-optimal according to given criteria (e.g., minimal number of terms*) imposes a natural order on the knowledge being

*i.e., products of selectors (Michalski, 1973). A simple form of a term is a conjunctive statement.

acquired. Another serious problem in Tan's program is the choice of adequate predicates. In a complicated end-game it will certainly not be easy to find the appropriate ones. Here again inductive learning could be used to generate predicates. A possibility for solving this particular problem is suggested in the end of the paper. Also, Tan's program is not a simple one in terms of the length and speed. A program to play an end-game using variable-valued logic formulas would be much simpler in those aspects because finding the value of a $VL_1$ formula for a specific event (chess position) is a very straightforward and simple operation.

The experiments described in the following sections represent the first step in testing these possibilities. It consists basically in generating variable-valued logic formulas for the single-pawn end game and testing the formulas. These were generated by inputting to AQ7 a set of learning events for each class of positions (winning or drawing). The formulas were tested by a different set of events. A testing event was assigned to one class if the percentage of terms that covered that event in the formula generated for that class was greater than the one in the formula obtained for the other class. In case of tie, the event was assigned to an undecided class.

### First experiment

In the first experiment a sample of 242 learning events was used (120 winning positions and 122 drawing positions). These events were generated by taking examples from the classes of positions defined by the terminal nodes of the decision tree of Tan's program. The events were described by the following set of 17 variables (the numbers between the parenthesis define the range of the variables):

- $x_1$ TURN(0:1): 0-BLACK; 1-WHITE.
- $x_2$ PAWN CAN ADVANCE(0:1): If equal to 1, the pawn can move without being captured in the next move. That also implies that the pawn is not blocked.
- $x_3$ PAWN's FILE(0:1): 0-not rook pawn; 1-rook pawn.
- $x_4$ WHITE KING BLOCKS PAWN(0:1): If equal to 1 the white king is in front of the pawn.
- $x_5$ WHITE STALEMATE(0:1): If equal to 1 the white have no legal move.
- $x_6$ CAN CAPTURE(0:1): If equal to 1 the black king can capture the pawn in the present move. This variable has no meaning if it is white's turn.
- $x_7$ BLACK STALEMATE(0:1): If equal to 1 the black king has no legal move.
- $x_8$ PAWN CAN RUN(0:1): If equal to 1 the pawn can keep moving till it gets crowned without being intercepted by the black king.
- $x_9$ BLACK KING ON THE CORNER(0:1): If equal to 1 the black king is on an upper corner of the board.

$x_{10}$ WHITE KING DOMINANT(0:1): If equal to 1 then we have that either of the 2 following cases occurs:

I. $5 \leqslant$ rank of pawn $\leqslant 7$ and rank of white king = rank of pawn + 1 and file of pawn $-1 \leqslant$ file of white king $\leqslant$ file of pawn + 1 and $2 \leqslant$ file of pawn $\leqslant 7$.

II. rank of white king = rank of pawn + 2 and file of pawn $- 1 \leqslant$ file of white king $\leqslant$ file of pawn + 1.

$x_{11}$ BLACK KING AHEAD(0:1): If equal to 1 the black king blocks the pawn.

$x_{12}$ PAWN'S RANK(0:3): 0-ranks 2, 3, and 4; 1-rank 5; 2-rank 6; 3-rank 7.

$x_{13}$ WHITE KING'S RANK(0:5): 0-2 rows behind the pawn; 1-1 row behind the pawn, . . .; 5-3 rows in front of the pawn.

$x_{14}$ WHITE KING'S FILE(0:3): 0-same as the pawn's file; 1-1 column from the pawn, . . .; 3-3 columns from the pawn.

$x_{15}$ BLACK KING'S RANK(0:4): 0-1 row behind the pawn; . . .; 4-3 rows in front of the pawn.

$x_{16}$ BLACK KING FILE'S(0:3): Similar to $x_{14}$.

$x_{17}$ RELATIVE POSITION OF THE KINGS(0:1): 0-the kings are on the same side of the pawn, 1-the kings are on opposite sides of the pawn.

The first 11 variables correspond to some predicates used by Tan. The other variables, called positional variables, define the coordinates of the men on the board (it has been assumed, however, that only positions in which the kings are no more than 3 moves away from the pawn can be represented).

AQ7 was run twice. The learning events were presented in a different order each time. In the first run the events in the winning class were covered by 16 terms and the events in the drawing class by 18 terms. For the second run these number were 14 and 23 respectively. The formulas obtained are listed in the Appendix. The great majority of rules determined by the formulas were heavily dependent on the positional variables. The rules can easily be expressed in natural language. Here are some examples:

> "Black can draw the game when the pawn is a rook pawn and the black king is ahead of the pawn on the same column or on the adjacent one."
>
> "Black can draw the game when it's Black's turn, the pawn cannot advance and the white king does not block the pawn." (Actually there is one exception to this rule and that is the case in which the rank of the pawn is 7, the black king blocks the pawn and the white king is on the square immediately behind the pawn on its left or right side.)
>
> "White wins when it is White's turn, the pawn has a rank 7 and is not

a rook pawn, and the white king is on the square immediately behind the pawn and on the same column."

The formulas were tested with a sample of 159 events* (78 drawing positions and 81 winning positions). The testing events were generated in an analogous way to the learning events. The following table shows for each class of positions, the percentages of events classified correctly, incorrectly and not classified, for each formula separately and for both formulas considered as only one.

| | Drawing Positions | | | Winning Positions | | |
|---|---|---|---|---|---|---|
| | Correct | Incorrect | Undecided | Correct | Incorrect | Undecided |
| First Formulas | 86% | 5% | 9% | 86% | 4% | 10% |
| Second Formulas | 83% | 12% | 5% | 79% | 12% | 9% |
| Both Formulas Together | 80% | 4% | 16% | 80% | 4% | 16% |

**Second experiment**

The second experiment consisted of two parts. In the first part, the formulas were obtained by running AQ7 with a set of 200 learning events† (97 drawing positions and 103 winning positions) randomly generated. In the second part, new formulas were generated by adding 54 additional events to the previous set of 200. These new 54 positions were obtained in the way described in the first experiment. The events were described by the following set of 15 variables (These variables are equal or similar to the variables used in the first experiment; variables WHITE STALEMATE or BLACK STALEMATE were not used here):

$x_1$ TURN(0:1): 0-BLACK, 1-WHITE. Same as $x_1$ in the first experiment.

$x_2$ PAWN'S RANK(0:5): 0-rank 2, . . ., 5-rank 7. The ranks 2, 3, and 4 that were clustered in the first experiment were separated here. Same as $x_{12}$ in the first experiment.

$x_3$ PAWN'S FILE(0:1): 0-not rook pawn, 1-rook pawn. Same as $x_3$ in the first experiment.

$x_4$ WHITE KING'S RANK(0:6): 0-3 rows behind the pawn, 1-2 rows behind the pawn, . . ., 6-3 rows in front of the pawn. The variable

*The total number of events possible (the cardinality of the cartesian product of the domains of descriptors is approximately $8 \cdot 10^6$). It should be noted that this number is much larger than the number of possible board situations for which the upper-bound is only 64 x 63 x 62 = 250,000. The discrepancy is due to the fact that certain combinations of descriptor values correspond to impossible chess board situations.

†The size of event space here is approximately $5 \cdot 10^6$.

$x_{13}$ of the first experiment was expanded to include 3 rows behind the pawn.

$x_5$ WHITE KING'S FILE(0:3): 0-same as the pawn's file, 1-1 column from the pawn, . . ., 3-3 columns from the pawn. Same as $x_{14}$ in the first experiment.

$x_6$ BLACK KING'S RANK(0:6): 0-3 rows behind the pawn, 1-2 rows behind the pawn, . . ., 6-3 rows in front of the pawn. The variable $x_{15}$ of the first experiment was expanded to include 3 rows behind the pawn.

$x_7$ BLACK KING'S FILE(0:3): 0-same as the pawn's file, 1-1 column from the pawn, . . ., 3-3 columns from the pawn. Same as $x_{16}$ in the first experiment.

$x_8$ RELATIVE POSITION OF THE KINGS(0:1): 0-the kings are on the same side of the pawn, 1-the kings are on opposite sides of the pawn. Same as $x_{17}$ in the first experiment.

$x_9$ BLACK KING ON THE CORNER(0:1): If equal to 1 the black king is on an upper corner of the board. Same as $x_9$ in the first experiment.

$x_{10}$ PAWN CAN ADVANCE(0:1): If equal to 1 the pawn can move without being captured in the next move. That also implies that the pawn is not blocked. Same as $x_2$ in the first experiment.

$x_{11}$ BLACK KING FAR FROM THE PAWN(0:1): If equal to 1 the black king is outside the region where it can intercept the pawn before it gets crowned by keeping moving. That does not mean that the pawn can run because it can be blocked by the white king. This variable is a generalization of the variable $x_8$ of the first experiment.

$x_{12}$ WHITE KING BLOCKS PAWN(0:1): If equal to 1 the white king is in front of the pawn. Same as $x_4$ in the first experiment.

$x_{13}$ BLACK KING IS AHEAD(0:1): If equal to 1 the black king is in front of the pawn. Same as $x_{11}$ in the first experiment.

$x_{14}$ WHITE KING IS ON CRITICAL SQUARES(0:1): If equal to 1 the position of the white king is defined by the following inequalities.

pawn's rank $<$ white king's rank $\leqslant$ pawn's rank + 2 pawn's file - 1 $\leqslant$ white king's file $\leqslant$ pawn's file + 1. This variable is a generalization of the variable $x_{10}$ of the first experiment.

$x_{15}$ PAWN UNPROTECTED(0:1): If equal to 1 than if it is black's turn the black kind can take the pawn in the next move. This variable is a generalization of the variable $x_6$ of the first experiment.

The formulas obtained for this experiment were again heavily dependent on the positional variables and are rather complicated. For the first part the events in the winning class were covered by 11 terms and the events on the drawing class were covered by 9 terms. For the second part these numbers were 23 and 23 respectively. The formulas are listed in the Appendix.

The formulas were tested with a sample of 200 events (86 drawing positions and 114 winning positions). The testing events were randomly generated. The results of the test are shown in the table below:

| Learning Events | Drawing Positions | | | Winning Positions | | |
|---|---|---|---|---|---|---|
| | Correct | Incorrect | Undecided | Correct | Incorrect | Undecided |
| 200 random events | 82% | 8% | 10% | 92% | 5% | 3% |
| 200 random events + 54 selected events | 85% | 6% | 9% | 83% | 4% | 13% |

## CONCLUSIONS AND SUGGESTIONS FOR FUTURE EXPERIMENTS

The results obtained in the two experiments described in this paper should be considered quite good if we realize that they represent the very first step in the learning process. The fact that the formulas obtained were heavily dependent on the lower-level positional variables seems to indicate that the other variables used (the ones corresponding to Tan's predicates) are not very significant in terms of the number of positions that they can cover. This was stressed in the second experiment when the events that are nicely described by these variables were added to the sample of randomly generated learning events and did not produce any significant improvement in the formulas. New variables should be introduced, probably ones corresponding to other predicates represented in Tan's program either through position patterns or, in a more complex way, through lookahead procedures that "test values of positions resulting from a hypothetical execution of an action scheme."

The next step in the chess-end-game experiment could be the improvement of the obtained formulas by feedback learning. This is a multi-step process that can be implemented in the following way:

1. Test the formulas with a sample of testing events. If all events are classified correctly stop, If not, go to step 2.
2. Obtain new formulas by inputting the formulas so far obtained to AQ9 program together with the events classified incorrectly in step 1. (AQ9 accepts events as well as terms as inputs.) Go to step 1.

This process should eventually produce quasi-correct formulas in the sense that all the events presented so far as inputs to the learning programs would be correctly classified by the formulas. The process can obviously be repeated ad nauseam with different samples of testing events. It is of great interest to see whether this process would eventually produce 'stable' decision rules and, also, how fast it could produce such rules, if it is possible to produce them in a reasonable time.

When dealing with more complicated end-games for which a nice set of descriptors (variables) is not available, it seems that some way of "discovering" new variables is necessary.

One possibility is to start generating formulas using exclusively lower-level variables (which describe only the positions of each man on the board). At the same time a set of position patterns is kept in the memory. These patterns could be described by some relations between the primitive variables, by array-images of the board, by geometric relations among some men on the board, or in any other convenient way. Every time that new learning events are input to AQVAL, some statistics are taken from them, so that each pattern will have associated with it the number of times that it occurred in each class of positions (winning, drawing, or losing). Whenever these numbers reach a certain threshold that shows that the corresponding pattern has some significance in separating the classes, the pattern is tentatively introduced as a new variable. If the use of that new variable simplifies the formulas so far obtained, it is kept with the other variables. If not, it is forgotten. These positional patterns can be at first introduced by humans. In a more advanced step they could be generated by the machine.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to Professor Donald Michie for the encouragement, useful comments, and interest in this work. This work was partially supported by the National Science Foundation under Grant No. DCR 74-03514.

## REFERENCES

Awerbach, J., (1958) *Lehrbuch der Endspiele*, 1, Bauernendspile, Berlin.

Chernev, I. (1961) *Practical Chess Endings*, (Dover Publications, Inc., New York).

Cuneo, R.P. (1975) Selected Problems of Minimization of Variable-Valued Logic Formulas, M.S. Thesis, Department of Computer Science, University of Illinois.

Fine, R. (1941) *Basic Chess Endings* (David McKay Co., Inc., New York).

Horowitz, I.A. (1973) *How to Win in Chess Endings*, (David McKay Co., Inc., New York).

Jensen, G.M. (1975) SYM-1: A Program that Detects Symmetry of Variable-Valued Logic Functions, Department of Computer Science, University of Illinois.

Larson, J. and R.S. Michalski (1975) AQVAL/1 (AQ7): User's Guide and Program Description, Department of Computer Science, University of Illinois.

Michalski, R.S. (1973) Discovering Classification Rules by the Variable-Valued Logic System $VL_1$, Proc IJCAI3, Stanford, California.

Michalski, R.S. (1973) AQVAL/1—Computer Implementation of a Variable-Valued Logic System and the Application to Pattern Recognition," *Proceedings of the First International Joint Conference on Pattern Recognition*, Washington, D.C., October 30-November 1, 1973.

Michalski, R.S. (1974) Learning by Inductive Inference, NATO Advanced Study Institute on Computer Oriented Learning Processes, August 26-September 7, 1974, France (invited paper).

Michalski, R.S. (1975) Variable-Valued Logic and Its Applications to Pattern Recognition and Machine Learning, in *Multiple-Valued Logic and Computer Science*, (North-Holland Publishers).

Michie, D. A Theory of Evaluative Comments in Chess, *Proceedings of the AISB Summer Conference*, Brighton, University of Sussex.

Michie, D., A.P. Ambler, and R. Ross, Memory Mechanisms and Machine Learning, in *Simple Nervous Systems* (ed. B.R. Newth and P.W.R. Usherwood, London, Edward Arnold).

Newborn, M. (1975) *Computer Chess* (Academic Press, New York).

Pitrat, J. (1974) Realization of a Program Learning to Find Combinations in Chess, in *Computer Oriented Learning Processes*, NATO Advanced Study Institute, August 27-September 6, Bonas (gers) 1974.

Tan, S.T. (1972) Representation of Knowledge for Very Simple Endings in Chess, *Memorandum MIP-R-98*, School of Artificial Intelligence, University of Edingburgh.

## APPENDIX

## LIST OF THE FORMULAS GENERATED

The formulas obtained from $VL_1$ for the experiments are given in the tables below. Each line of a table represents a term of a $VL_1$ formula; each table is a complete formula, the disjunction of its terms.

Four values are tabulated for each term:

COV – Number of learning events covered by the term

NEW – Number of learning events covered by the term that were not covered by previous terms

IND – Number of learning events covered exclusively by this term

TOT – Number of learning events covered by all terms listed so far

| $VL_1$ Formula | COV | NEW | IND | TOT |
|---|---|---|---|---|
| $[x_8=0]\ [x_{11}=0]\ [x_{12}=0,3]\ [x_{13}=0:3]\ [x_{14}=1,3]\ [x_{15}=1,3]\ [x_{16}=0,1]$ ∨ | 19 | 19 | 4 | 19 |
| $[x_8=0]\ [x_{11}=0]\ [x_{12}=0,1,3]\ [x_{13}=0,1]\ [x_{15}=1,3,4]\ [x_{16}=0,1]\ [x_{17}=0]$ ∨ | 11 | 5 | 3 | 24 |
| $[x_8=0]\ [x_{13}=1:3]\ [x_{14}=1,3]\ [x_{15}=2,3]\ [x_{16}=0]$ ∨ | 24 | 20 | 4 | 44 |
| $[x_4=0]\ [x_8=0]\ [x_{12}=3]\ [x_{13}=0,3]\ [x_{14}=3]\ [x_{15}=0,2]$ ∨ | 3 | 2 | 1 | 46 |
| $[x_1=0]\ [x_4=0]\ [x_8=0]\ [x_{13}=0,1,4]\ [x_{15}=2:4]\ [x_{16}=0,1]$ ∨ | 39 | 27 | 3 | 73 |
| $[x_1=0]\ [x_4=0]\ [x_8=0]\ [x_{12}=0,3]\ [x_{13}=3]\ [x_{15}=1,4]$ ∨ | 3 | 3 | 0 | 76 |
| $[x_3=1]\ [x_8=0]\ [x_{15}=2:4]\ [x_{16}=o,1]\ [x_{17}=0]$ ∨ | 43 | 17 | 13 | 93 |
| $[x_{10}=0]\ [x_{11}=0]\ [x_{12}=0:3]\ [x_{13}=2,3]\ [x_{15}=1:3]\ [x_{16}=0:2]\ [x_{17}=0]$ ∨ | 18 | 7 | 4 | 100 |
| $[x_3=1]\ [x_8=0]\ [x_{10}=0]\ [x_{12}=1,2]\ [x_{14}=0,2,3]\ [x_{15}=0,1]\ [x_{16}=0,1]\ [x_{17}=0]$ ∨ | 4 | 4 | 3 | 104 |
| $[x_1=1]\ [x_3=0]\ [x_{12}=2]\ [x_{13}=2]\ [x_{14}=1]\ [x_{15}=3]\ [x_{16}=1]\ [x_{17}=0]$ ∨ | 2 | 2 | 2 | 106 |
| $[x_1=0]\ [x_{12}=2]\ [x_{13}=3]\ [x_{14}=1]\ [x_{15}=3]\ [x_{16}=1]$ ∨ | 1 | 1 | 1 | 107 |
| $[x_{10}=0]\ [x_{12}=0,2]\ [x_{13}=1,3,4]\ [x_{14}=0]\ [x_{15}=2:4]\ [x_{16}=0,2]$ ∨ | 14 | 6 | 6 | 113 |
| $[x_1=0]\ [x_4=0]\ [x_8=0]\ [x_{14}=1]\ [x_{15}=0,3]\ [x_{16}=0,2]$ ∨ | 13 | 3 | 2 | 116 |
| $[x_1=0]\ [x_3=1]\ [x_4=0]\ [x_8=0]\ [x_{12}=2]\ [x_{13}=2]$ ∨ | 5 | 1 | 1 | 117 |
| $[x_4=0]\ [x_8=0]\ [x_{12}=0]\ [x_{13}=3]\ [x_{14}=1]\ [x_{15}=4]\ [x_{16}=1]$ ∨ | 1 | 1 | 1 | 118 |
| $[x_2=0]\ [x_{13}=3]\ [x_{14}=2]\ [x_{15}=2]$ ∨ | 2 | 1 | 1 | 119 |
| $[x_1=0]\ [x_2=0]\ [x_4=0]$ ∨ | 43 | 2 | 2 | 121 |
| $[x_2=0]\ [x_3=0]\ [x_{12}=2]\ [x_{13}=4]\ [x_{14}=2]\ [x_{15}=3]\ [x_{17}=1]$ | 1 | 1 | 1 | 122 |

TABLE A1. First experiment, first run, drawing class

| $VL_1$ Formula | COV | NEW | IND | TOT |
|---|---|---|---|---|
| $[x_3=0]\ [x_6=0]\ [x_{13}=1:4]\ [x_{14}=0:2]\ [x_{15}=0:2]$ V | 51 | 51 | 22 | 51 |
| $[x_1=1]\ [x_3=0]\ [x_9=0]\ [x_{11}=0]\ [x_{12}=2]\ [x_{13}=1,3\ 4]\ [x_{15}=3]\ [x_{16}=1:3]$ V | 13 | 13 | 10 | 64 |
| $[x_1=1]\ [x_{11}=0]\ [x_{12}=2,3]\ [x_{14}=1:3]\ [x_{16}=2,3]$ V | 18 | 12 | 7 | 76 |
| $[x_1=1]\ [x_5=0]\ [x_{11}=0]\ [x_{12}=0,1,3]\ [x_{13}=2:4]\ [x_{14}=0,2]\ [x_{15}=0,1,4]\ [x_{16}=1:3]$ V | 21 | 9 | 5 | 85 |
| $[x_1=0]\ [x_2=1]\ [x_{12}=2]\ [x_{13}=2]\ [x_{16}=1,3]$ V | 5 | 3 | 3 | 88 |
| $[x_6=0]\ [x_{11}=0]\ [x_{12}=1,2]\ [x_{13}=2:4]\ [x_{15}=0,4]\ [x_{16}=1,2]$ V | 14 | 9 | 8 | 97 |
| $[x_1=1]\ [x_{13}=0,2]\ [x_{14}=0,1]\ [x_{15}=2,4]$ V | 7 | 5 | 5 | 102 |
| $[x_1=1]\ [x_5=0]\ [x_{12}=1:3]\ [x_{13}=3,4]\ [x_{15}=0,1,4]\ [x_{16}=0,2]$ V | 15 | 6 | 5 | 108 |
| $[x_1=1]\ [x_6=0]\ [x_{13}=4,5]\ [x_{14}=1,3]\ [x_{15}=0,1]\ [x_{16}=1]$ V | 4 | 2 | 2 | 110 |
| $[x_3=0]\ [x_{12}=2,3]\ [x_{13}=0,2]\ [x_{14}=2]\ [x_{15}=2,3]$ V | 7 | 5 | 4 | 115 |
| $[x_2=1]\ [x_3=0]\ [x_{12}=0]\ [x_{13}=3]\ [x_{16}=0]$ V | 1 | 1 | 1 | 116 |
| $[x_1=1]\ [x_3=0]\ [x_{12}=3]\ [x_{13}=1]\ [x_{14}=0]$ V | 1 | 1 | 1 | 117 |
| $[x_1=1]\ [x_{12}=2]\ [x_{13}=2]\ [x_{17}=1]$ V | 4 | 1 | 1 | 118 |
| $[x_4=1]\ [x_{12}=1,3]\ [x_{16}=0,3]$ | 9 | 2 | 2 | 120 |

TABLE A2. First experiment, first run, winning class

| $VL_1$ Formula | COV | NEW | IND | TOT |
|---|---|---|---|---|
| $[x_3=1]\ [x_{10}=0]\ [x_{13}=0,1,2,5]\ [x_{14}=0,1,3]\ [x_{15}=0:3]\ [x_{16}=0,1]\ [x_{17}=0]$ ∨ | 30 | 30 | 4 | 30 |
| $[x_1=1]\ [x_{10}=0]\ [x_{12}=0,2,3]\ [x_{13}=2,3]\ [x_{14}=1,3]\ [x_{15}=1:3]\ [x_{16}=0,1]\ [x_{17}=0]$ ∨ | 10 | 8 | 2 | 38 |
| $[x_1=0]\ [x_{10}=0]\ [x_{13}=0,1,4]\ [x_{14}=0,3]\ [x_{15}=2,3]$ ∨ | 16 | 8 | 1 | 46 |
| $[x_2=0]\ [x_3=1]\ [x_{10}=0]\ [x_{13}=2,3]\ [x_{14}=0,2]\ [x_{15}=2]\ [x_{16}=0,2]$ ∨ | 8 | 8 | 4 | 54 |
| $[x_2=0]\ [x_{10}=0]\ [x_{12}=0,3]\ [x_{13}=1:3]\ [x_{14}=1,3]$ ∨ | 20 | 10 | 6 | 64 |
| $[x_4=0]\ [x_{12}=0:2]\ [x_{13}=2:4]\ [x_{14}=1,2]\ [x_{15}=2,3]\ [x_{16}=0]\ [x_{17}=0]$ ∨ | 16 | 9 | 2 | 73 |
| $[x_{10}=0]\ [x_{12}=2]\ [x_{13}=4]\ [x_{14}=0]\ [x_{15}=2]\ [x_{16}=2]\ [x_{17}=0]$ ∨ | 1 | 1 | 1 | 74 |
| $[x_3=1]\ [x_4=0]\ [x_8=0]\ [x_{10}=0]\ [x_{13}=1,4]\ [x_{14}=0,2]\ [x_{15}=0,1,2,4]\ [x_{16}=0,1]\ [x_{17}=0]$ ∨ | 11 | 5 | 1 | 79 |
| $[x_4=0]\ [x_{10}=0]\ [x_{12}=0,3]\ [x_{13}=0,3]\ [x_{15}=1,4]\ [x_{16}=1]\ [x_{17}=0]$ ∨ | 9 | 6 | 3 | 85 |
| $[x_2=0]\ [x_{10}=0]\ [x_{12}=0,2]\ [x_{15}=2:4]\ [x_{16}=0]\ [x_{17}=0]$ ∨ | 29 | 7 | 5 | 92 |
| $[x_1=0]\ [x_{13}=1,3,4]\ [x_{15}=3]$ ∨ | 18 | 6 | 6 | 98 |
| $[x_1=0]\ [x_{10}=0]\ [x_{12}=0,3]$ ∨ | 24 | 6 | 3 | 104 |
| $[x_1=0]\ [x_3=0]\ [x_{12}=0,1]\ [x_{13}=2,4]\ [x_{14}=1]\ [x_{15}=0,3]\ [x_{16}=0,1]$ ∨ | 5 | 2 | 2 | 106 |
| $[x_2=0]\ [x_{10}=0]\ [x_{13}=0]\ [x_{14}=3]\ [x_{15}=2]$ ∨ | 1 | 1 | 1 | 107 |
| $[x_{10}=1]\ [x_{12}=3]\ [x_{13}=3]\ [x_{15}=1,2]\ [x_{16}=0,2]$ ∨ | 4 | 2 | 2 | 109 |
| $[x_3=1]\ [x_{10}=0]\ [x_{14}=0]\ [x_{15}=3]$ ∨ | 6 | 2 | 2 | 111 |
| $[x_2=0]\ [x_{10}=0]\ [x_{13}=4]\ [x_{14}=2]\ [x_{15}=3]\ [x_{17}=1]$ ∨ | 1 | 1 | 1 | 112 |
| $[x_1=0]\ [x_{10}=0]\ [x_{14}=1]\ [x_{15}=3]\ [x_{16}=2]$ ∨ | 2 | 2 | 2 | 114 |
| $[x_1=0]\ [x_3=1]\ [x_{14}=2]\ [x_{15}=3,4]$ ∨ | 5 | 3 | 3 | 117 |
| $[x_2=1]\ [x_{13}=1]\ [x_{14}=1]\ [x_{15}=4]$ ∨ | 2 | 2 | 2 | 119 |
| $[x_2=0]\ [x_{13}=0]\ [x_{14}=3]\ [x_{15}=1]$ ∨ | 2 | 1 | 1 | 120 |
| $[x_1=0]\ [x_2=0]\ [x_4=0]\ [x_{13}=3]$ ∨ | 4 | 1 | 1 | 121 |
| $[x_1=1]\ [x_3=1]\ [x_4=0]\ [x_{14}=2]\ [x_{16}=1]$ | 3 | 1 | 1 | 122 |

TABLE A3. First experiment, second run, drawing class