Structured and Unstructured Induction with EDAGs

Brian R. Gaines
Knowledge Science Institute
University of Calgary
Alberta, Canada T2N 1N4

Abstract

One objective of knowledge acquisition using inductive modeling has been the development of correct and effectively computable descriptions that can also be assimilated and used by a human being. On substantial problems neither trees nor rules satisfy this objective, both being generally difficult to understand as overt knowledge. Exception directed acyclic graphs (EDAGs) are knowledge structures that subsume trees and rules but can be substantially more compact. The comprehensibility of manually constructed and induced EDAGs is investigated by reconstructing Shapiro's "structured induction" of a solution to the pawn versus rook end game. It is shown that the induced EDAG is very similar to that produced through consultation with experts, and that both are small, comprehensible solutions to a problem that is difficult for people.

1 Introduction

One expectation of knowledge acquisition using inductive modeling of representative cases has been that it may produce a model that is not only effective but also comprehensible and insightful to an expert. Michalsi and Chilausky's (1980) study of soybean pathology diagnosis not only produced a model that was more accurate than that directly elicited from the expert but also one that included rules that the expert understood and decided to use in teaching. However, such experience has not been widely replicated, and the models produced by common inductive modeling programs such as ID3 and C4.5 (Quinlan, 1993) generally consist of a large decision tree or collection of rules that are not comprehensible to people even if they are effective in problem solving. Quinlan (1991) suggests that such models cannot be considered as "knowledge":

> *"Donald Michie (1986) identifies* concept expressions *as those correct and effectively computable descriptions that can also be assimilated and used by a human being. As a counterexample, he cites a case in which ID3 derived a decision tree for a chess end game from a complete set of positions. The tree was absolutely correct and computationally efficient but, alas, completely incomprehensible to human chess experts. As he put it, "It was not a question of a few glimmers of sense here and there scattered through a large obscure structure, but just a total blackout." In Michie's view, which I share, such a structure does not qualify as knowledge."*

The efficiency of existing techniques for generating compact trees and sets of rules suggests that it is not the modeling methodology which is at fault but rather the representational schema induced. A variety of generalizations of trees and rules have been proposed that offer the possibility of more compact and hence, hopefully, more comprehensible models. Gaines' (1989) *Induct* induces rule graphs that cover common cases by default rules and infrequent cases through linked exception rules. Compton and Jansen's (1990) *ripple-down rules* generalize binary decision trees by allowing a node to contain a compound premise, and interior nodes to contain conclusions which are asserted if the tree cannot be traversed further. Gaines (1991) shows that ripple-down rules are a particular case of *rules with exceptions* that can encode some knowledge structures more compactly. Oliver (1993) and Kohavi (1994) have shown how various forms of *decision graphs* may be induced and provide a more compact alternative than decision trees. Gaines (1995) generalizes these representations into *exception directed acyclic graphs* (EDAGs) that support exceptions within a general decision graph, and subsumes trees and rules. He shows that EDAGs induced for some simple standard datasets are substantially smaller and simpler than the equivalent trees and rules, and suggests that they are also more comprehensible.

However, none of the studies cited shows that the representational schema defined scales up to support the inductive modeling of large complex datasets in terms of a comprehensible knowledge structure. A problem in doing this is the lack of substantial datasets with well-defined expert models that allow inductive modeling to be evaluated in terms of humanly meaningful results. Shapiro’s (1987) study of the pawn versus rook end-game when the pawn is about to queen provides a complex dataset which has been modeled by a combination of human knowledge elicitation and inductive modeling. The main section of this article compares the model obtained direct induction of an EDAG for this problem with that obtained from human chess experts. Before this is done, the following section illustrates the induction of EDAGs for a simple chess problem.

2 Modeling a Simple Chess Dataset

Quinlan (1979) describes ID3 models of 7 rook versus knight end game situations of increasing difficulty. The third problem involves 647 cases with 4 3-valued attributes, 3 2-valued attributes, and a 2-valued outcome. Figure 1 shows the decision tree induced by ID3 that solves this problem graphed as an EDAG in Induct. Figure 2 shows the C4.5 rules derived from this tree graphed as an EDAG.

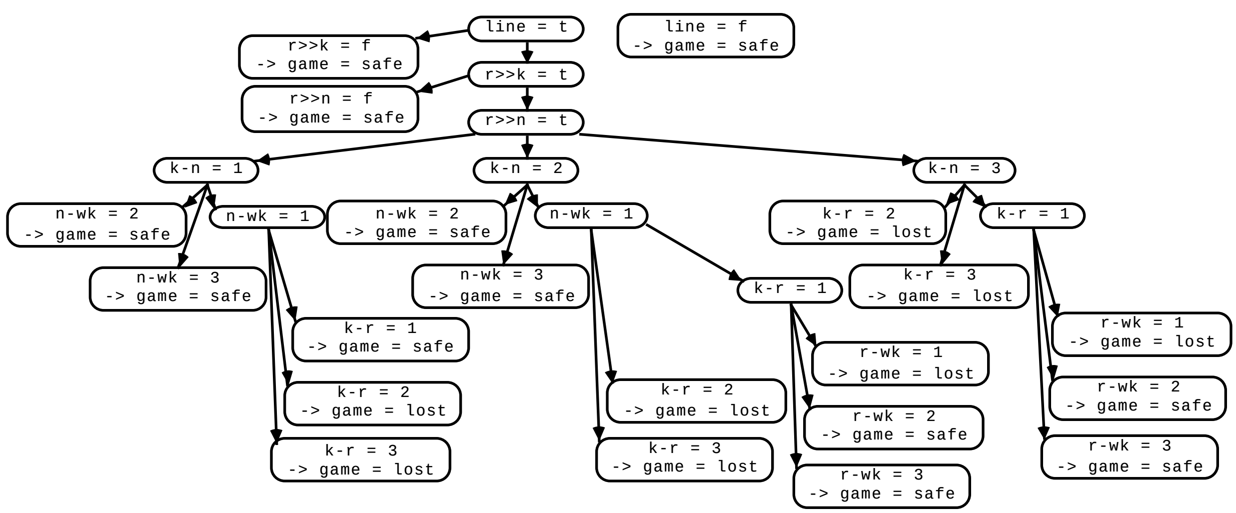

Figure 1 ID3 decision tree solving a rook versus knight chess end game problem

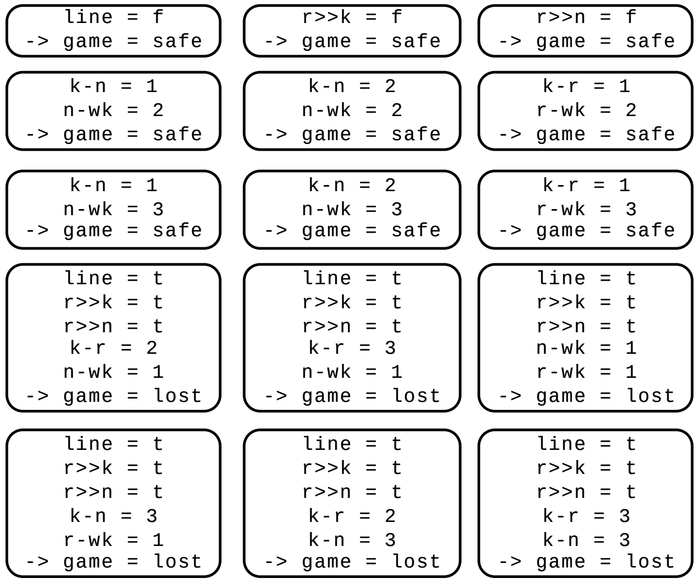

Figure 2 C4.5 rules derived from the decision tree of Figure 1

Both EDAGs are used for decision making through the same procedure. The graph is traced from its root nodes. On any path through the graph, if the premise (if any) is true then the conclusion (if any) is noted for that path, replacing any previous conclusion noted for the path. A path is traced until a premise fails to hold, or a leaf node is reached. When a path has been traced, any conclusion noted is asserted.

This procedure is trivially applicable to decision trees and rules. It also applies to general graphs of rules with exceptions. Figure 3 shows the rules of Figure 2 with the conclusion "game = safe" taken as a default and common clauses in the premises factored by Induct. The numbers indicate that 613 cases are covered by the default, and 34 are exceptions. Figure 4 shows a solution with multi-level exceptions directly induced by Induct. The numbers indicate that the exceptions may themselves be represented as a default covering 30 cases, an exception to that covering 10 cases, and an exception to that covering 4 cases.

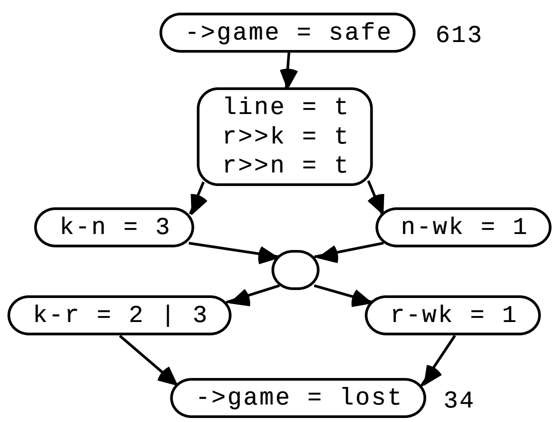

Figure 3 Induct factorization of the rules of Figure 2

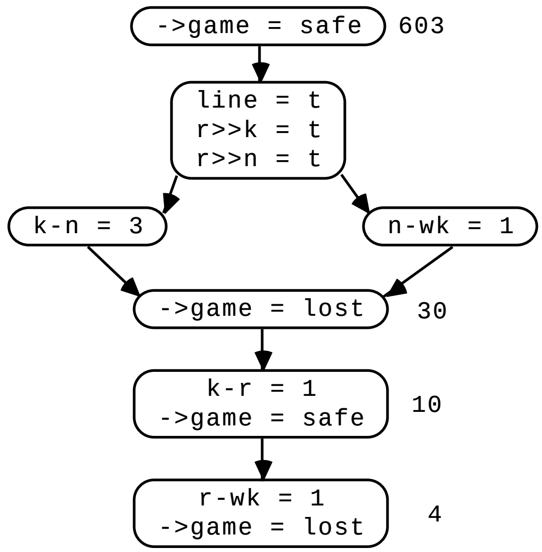

Figure 4 Induct rules with exceptions for the chess problem

The EDAGs of Figures 3 and 4 are logically equivalent to those of Figures 1 and 2 but substantially smaller and hence easier to understand. It would be very tedious to explain the decision tree or rules in words. However, the small EDAG in Figure 3 may be explicated as "the game is safe unless the black king, rook and knight are in line, the rook bears on the king, the rook bears on the knight, and either the king to knight distance is 3, or the knight to white king distance is 1, and either the king to rook distance is 2 or 3, or the rook to white king distance is 1." The EDAG in Figure 4 may be explicated as "the game is safe unless the black king, rook and knight are in line, the rook bears on the king, the rook bears on the knight, and either the king to knight distance is 3, or the knight to white king distance is 1, when the game is lost, unless the king to rook distance is 1, when it is won, unless the rook to white king distance is 1, when it is lost."

In both cases the graphic representation in the EDAGs seems more perspicuous than the text. EDAGs retain the flow of decision making that makes decision trees attractive. In addition, the reduced restrictions of the directed acyclic graph structure enables them to avoid some of the unnecessarily complex structures generated by redundant replication of nodes in a decision tree representing a disjunctive structure.

3 Comparing Human-Elicited and Induced Structures

How do the results of the previous section scale up to a more complex situation? Shapiro (1987) studied a situation in the rook versus pawn (KPa7KR) end-game situation which he notes is poorly documented in the chess literature. The white pawn is at a7, potentially able to queen in one move, and the black rook and the two kings are in any of the 209,718 legal configurations of which 129,825 are classified won-for-white and the remaining 79,893 are classified not-won-for-white.

In the chess situations studied by Quinlan and Shapiro the board positions of the pieces are well-defined but the attributes that a chess expert uses in assessing those positions are not. Hence, human experts were interviewed to elicit those characteristics of the situations that they saw as significant to making a decision about whether the game was won or not. Quinlan's studies elicited these attributes, applied them to a set of situations where the outcome was known, and then used ID3 to develop a decision tree which would be totally correct if the attributes adequately characterized the problem.

Shapiro's study addressed the problem noted in the initial quotation, that the resultant decision trees, although correct, could not be regarded as demonstrating "knowledge" of chess. He had chess experts partition the problem situation and define small sub-problems, each of which defined a new decision variable. He defined "small" to mean that the sub-problem could be solved using fewer than 7 attributes, each of which could be derived from a board position with less than a "screenfull of C code." For each sub-problem, he classified the database of cases in terms of the small set of attributes and the decision variable, and then used ID3 to develop a decision tree.

Shapiro termed this technique "structured induction" and used it to develop a complete solution of the KPa7KR end-game in terms of 9 sub-problems introducing 8 additional decision variables. The total number of attributes defined is 35 binary and 1 ternary, and the total number of different attribute vectors involved is 3,196. This dataset is available through the University of Irvine archive (Murphy and Aha, 1994), and has been used in other studies (Muggleton, 1987; Holte, Acker and Porter, 1989).

3.1 Structured Induction using Decision Trees

Figure 5 is reproduced from Shapiro's book and shows the structure of his solution in terms of the attributes used to develop the 9 sub-trees. Each circle represents a decision tree with between 4 and 7 attributes, and the complete structure has 122 nodes.

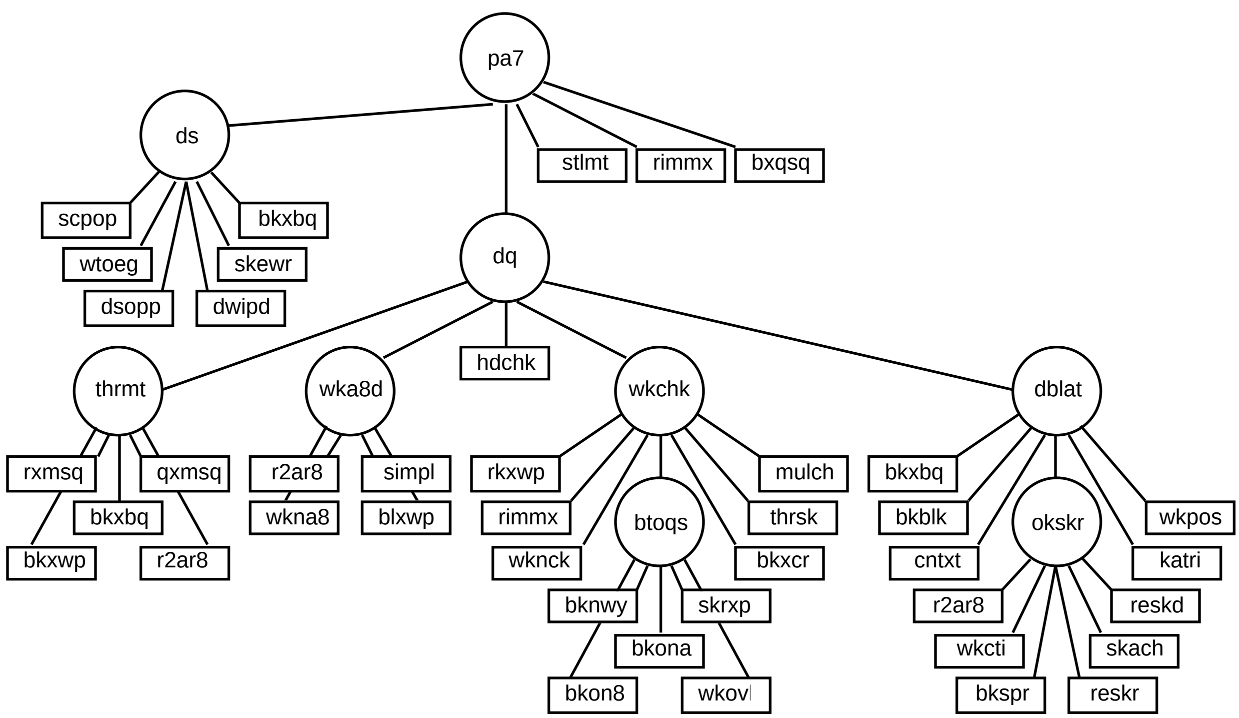

Figure 5 Structured induction of solution to a chess end game (Shapiro, 1987)

Figure 6 shows the top-level decision tree, pa7 in Figure 5. It involves the introduction of intermediate decision variables, such as ds (good delayed skewer threat) and dq (simple delay to queening), which are computed through other decision trees.

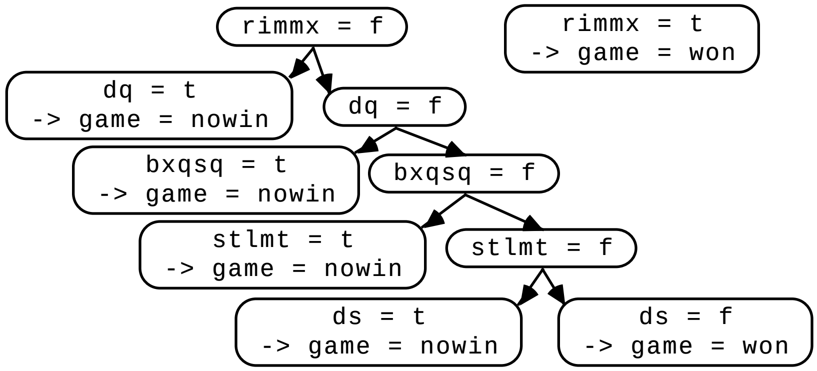

Figure 6 Top-level tree pa7 (Shapiro, 1987)

Figure 7 shows the decision tree for computing ds.

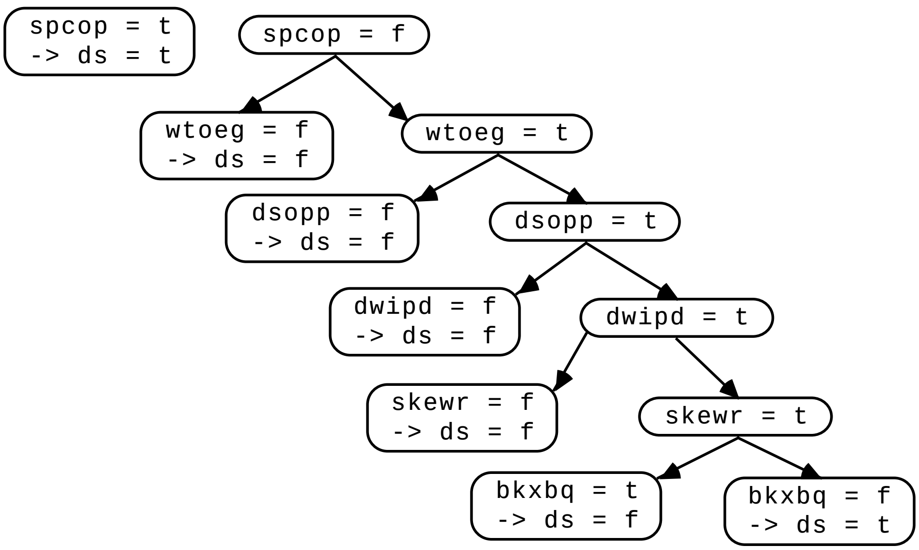

Figure 7 Sub-tree to determine ds (Shapiro, 1987)

Figure 8 shows the complete structured induction solution developed as a single EDAG by combining the trees to eliminate the intermediate decision variables. Context boxes have been placed around each sub-tree to retain the partition into sub-problems.

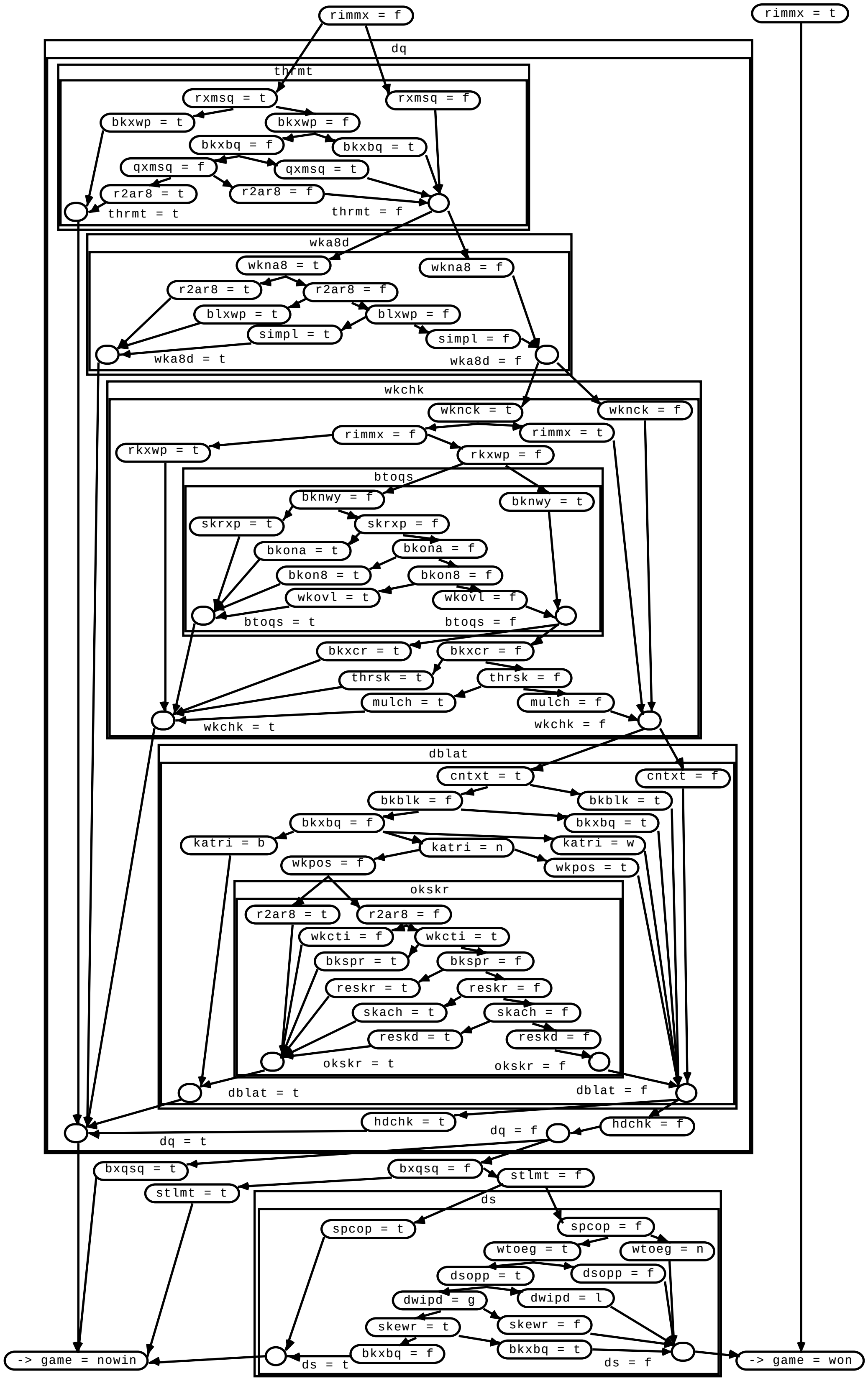

Figure 8 Structured induction trees combined by elimination of intermediate variables

3.2 Reconstruction of Structured Induction using EDAGs

Shapiro's use of decision trees reflects the tools and representational schema available to him in the early 1980s. The 9 sub-problem trees are themselves fairly difficult to understand, and are incomprehensible when combined into a single knowledge structure. As a first experiment ID3 was replaced with Induct and an EDAG induced for each sub-problem. Figure 9 shows the 9 EDAGs induced for the sub-problems. Enclosures have been added to show the sub-problems and their relationships. A nested EDAG computes an intermediate decision variable used by the EDAG at the level of nesting above.

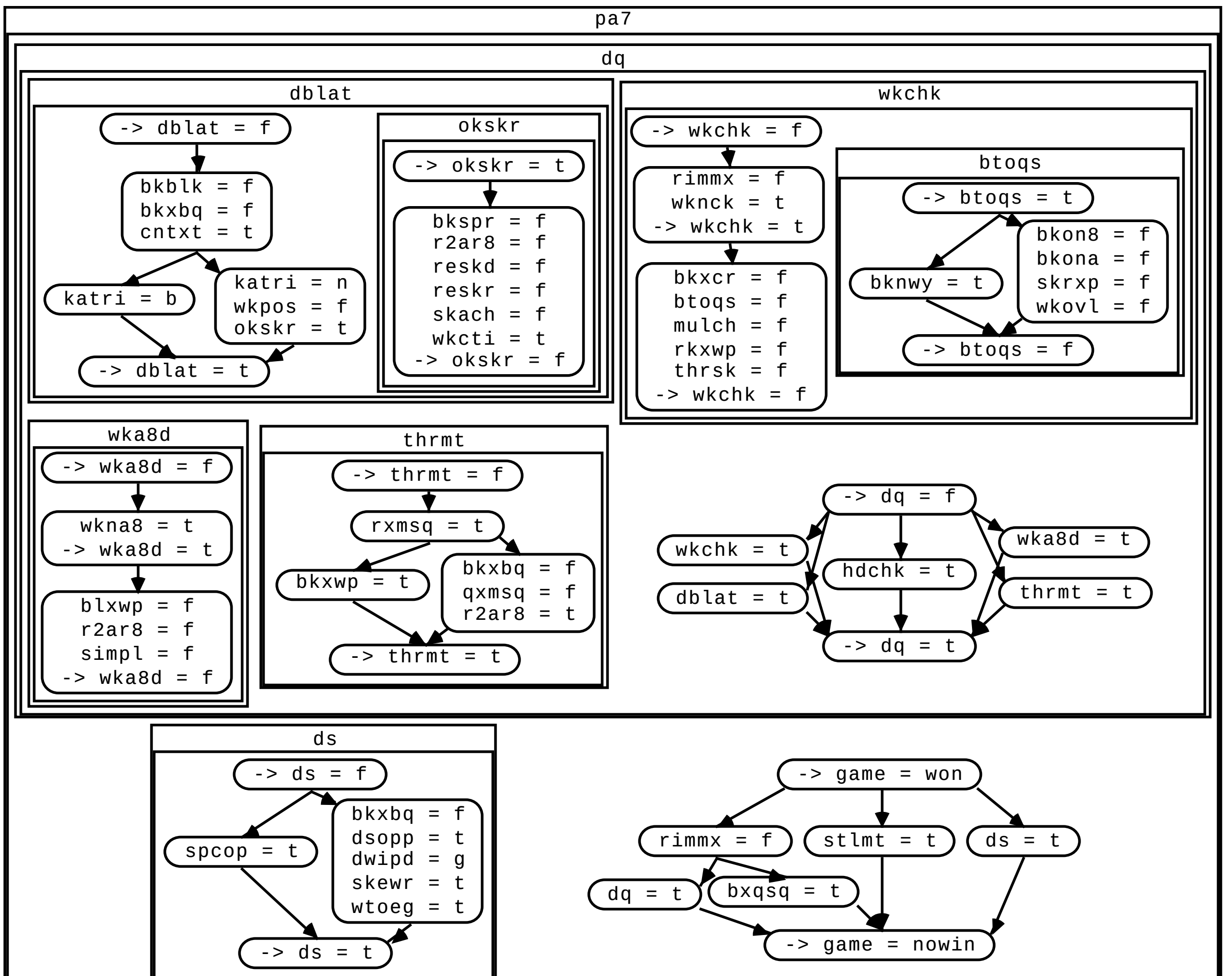

Figure 9 EDAGs induced for each of Shapiro's 9 sub-problems

It can be seen at the bottom right that the game is won unless the black rook cannot be captured safely (rimmx = f) and either there is a simple delay to queening (dq = t) or one or more black pieces controls the queening square (bxqsq = t), or the white king is in stalemate (stlmt = t), or there is a good delayed skewer threat (ds = t).

It can be seen at the bottom left that there is a good delayed skewer threat (ds = t) if there is a special opposition pattern (spcop = t), or a combination of 5 other attributes have the values specified. Each of the 8 intermediate decision variables is determined by a relatively simple and comprehensible EDAG. Those for wka8d (white king on promotion square) and wkchk (white king in check) are interesting because Induct has generated two-level exceptions for them.

Thus, the reconstruction of Shapiro's results in terms of EDAG's confirms more clearly than did the original ID3 trees that the problem has been structured into simple and sub-problems. The sub-problems are also meaningful because they correspond to situations defined by the expert in terms comprehensible to a chess player.

Figure 10 shows the EDAGs of Figure 9 converted to a single EDAG by elimination of the intermediate decision variables. This is simple if the EDAG computing the variable has a single exception which is the truth value specified when the variable is used. There is one sub-problem, okskr, where the exception is single but the exception is the opposite to that required. The variable is eliminated by treating okskr = f as an exception in the EDAG for dblat. This involves adding two additional clauses (mulch = f, bkxcr = f) which correspond to alternative paths to nowin that the simple exception does not take into account. Similarly, the elimination of the wkna8 and wknck variables computed through multi-level exceptions requires the introduction of the additional variables marked by bullets.

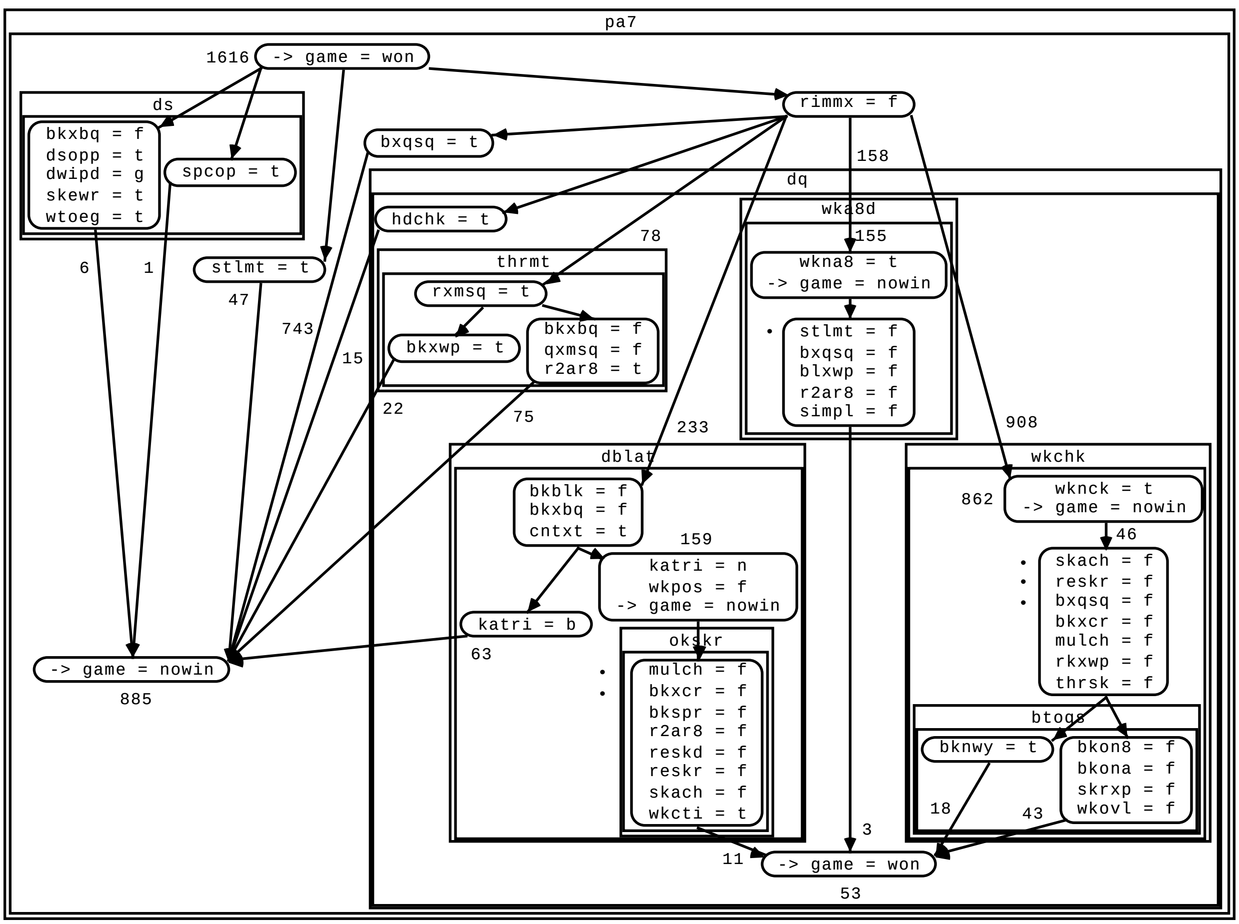

Figure 10 EDAG obtained from Figure 9 by eliminating the intermediate variables

Figure 10 is a single EDAG developed through structured induction that solves the KPa7KR problem. The numbers on the EDAG show the numbers of cases out of the total of 3196 that are covered by each part of the EDAG. It can be seen that the top level default accounts for 1616 cases, the path through scpop = t for 1 case, and so on. The numbers decided through all paths total substantially more than 3196 because a single case may flow through more than one path in an EDAG.

3.3 Comparing a Simply Induced Solution with that of Structured Induction

The EDAG of Figure 10 is a relatively simple exposition of the solution to what is said to be a difficult chess end- game situation, and it may be seen as a possible example of what Michie and Quinlan were seeking, a correct and effectively computable descriptions that can be assimilated and used by a human being. However, it was developed through a mixture of sub-problem structuring by human experts and inductive modeling. It is interesting to investigate whether a similar solution can be developed through inductive modeling without such human structuring.

Figure 11 shows the solution induced by Induct directly from the dataset of 3,196 cases. One difference from the EDAG of Figure 10 is that three of the attributes, bkxwp, rkxwp and stlmt, have not been used. Of these only stlmt seems to play a major role in Figure 10 accounting for 47 cases. However, if this path is removed only 2 errors occur so that 45 of the cases are covered by alternative paths. These 2 cases are covered by the rule wkna8 = t -> nowin in the EDAG of Figure 11.

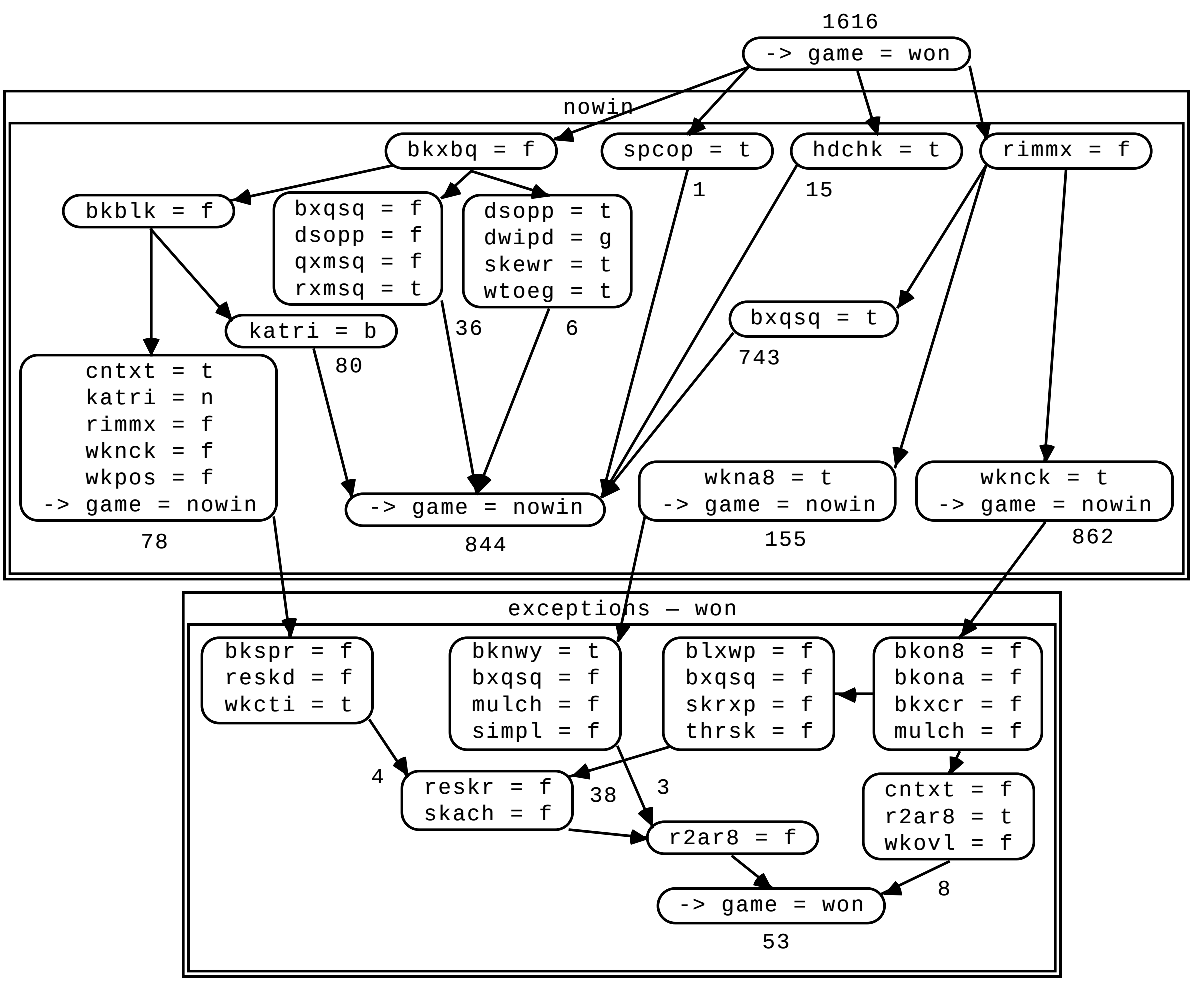

Figure 11 EDAG induced directly by Induct

Apart from the path through stlmt = t, it can be seen the top level structure above the exceptions in Figure 11 largely reproduces that of Figure 10. One difference is that some structures such as hdchk and dblat are under rimmx = f in Figure 10 and not in Figure 11. It turns out that it makes no difference to insert rimmx = f in these paths in Figure 11. It is a redundant clause that Induct eliminates.

Another difference is that the additional clause, wknck = f, has been introduced into the compound clause in ds. This corresponds to considering the delayed skewer threat only when the white king is not in check which is reasonable. It seems likely that the chess experts would have intended this when defining the skewer situation, but it was not taken into account when generating the cases for ID3.

The 53 exceptions to the top level structure are not generated through the same structures in Figures 10 and 11. The original okskr exception in dblat is reproduced exactly, but the exceptions relating to wkna8 and wknck are only loosely related in the two EDAGs. One possible reason for this is that there are two few cases for induction to determine the relevant attributes. In addition, these fairly rare and complex exception situations were simplified somewhat artificially in the Shapiro study using the ≤7 attributes and ≤1 screen of C criteria. They do not so closely correspond to basic chess concepts as do the upper level attributes. These exceptions and the differing knowledge structures generated need to be examined in detail by chess experts to determine precisely what is occurring.

One conclusion that it is reasonable to draw through comparison of Figures 10 and 11, is that the unguided induction of Induct has generated a knowledge structure of similar simplicity to the guided induction of Shapiro’s “structured induction”. Direct induction of an EDAG has also produced a correct and effectively computable description that can be assimilated and used by a human being.

4 Discusssion and Conclusions

The objective of producing inductive models that correspond to those of human experts is made difficult by the lack of test case data where expert models are available. Shapiro’s work provides an excellent, but unfortunately rare, opportunity to investigate the induction of comprehensible knowledge structures.

Muggleton (1987) was the first to take this opportunity in a reconstruction of Shapiro’s results using Duce, a system for constructive induction under manual control. Duce generated a correct and structured solution in a short time compared with Shapiro’s original studies. The detailed solution was not published but Duce generated 13 sub-problems involving 553 rules which is at least an order of magnitude larger than the solutions given in Figure 10 and 11.

The strong similarity of the solutions induced directly from the dataset and generated through the expert partitioning emphasizes the role of the expert in construing a problem. One way of interpreting the results is to say that the expert knowledge was primarily in the attributes specified and not so much in the way that the problem was structured. One could say that it is precisely the attributes specified by the experts that structure the problem. It will be noted that different attributes are used in the different sub-problems apparent in Figure 11. One could tell which sub-problem was relevant just by asking the expert which attributes he or she was considering.

Thus, even though the expert was absent in the development of the model of Figure 11, the knowledge of the expert was still present through the attributes used. The step from raw chess board positional data to the representation used in these studies is a large one, and Induct did not "learn to solve" the problem as a totally naive player would have to do. On the other hand, it seems likely that a totally naive human player would not be able to solve this problem anyway. A realistic human analogy would be a player who already knows concepts such as skewer and check, reads chess end-game books for other relevant concepts, and discusses situations with other players. In these terms one can say that such "background knowledge" of how to perceive a game was adequate for Induct to "learn to solve" the problem.

In conclusion, this study indicates that the EDAG knowledge structure that has previously been shown to subsume trees and decision rules while allowing more concise representations does scale up to a significant larger problem. It also shows that the knowledge structure produced by automatic induction of an EDAG can be very similar to that elicited directly from an expert. Clearly, many similar studies are necessary to substantiate these conclusions more generally, but the lack of datasets with expert associated expert models will make such studies slow to occur. Meanwhile EDAGs are of interest as potentially attractive alternatives to decision trees and rules.

Acknowledgements

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada.

References

Compton, P. and Jansen, R. (1990). A philosophical basis for knowledge acquisition. Knowledge Acquisition 2(3) 241-258.

Gaines, B.R. (1989). An ounce of knowledge is worth a ton of data: quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. Proceedings of the Sixth International Workshop on Machine Learning. pp.156-159. San Mateo, California, Morgan Kaufmann.

Gaines, B.R. (1991). Integrating rules in term subsumption knowledge representation servers. AAAI'91: Proceedings of the Ninth National Conference on Artificial Intelligence. pp.458-463. Menlo Park, California, AAAI Press/MIT Press.

Gaines, B.R. (1995). Transforming rules and trees into comprehensible knowledge structures. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R., Ed. Knowledge Discovery in Databases II. pp.205-226. Cambridge, Massachusetts, AAAI/MIT Press.

Holte, R.C., Acker, L.E. and Porter, B.W. (1989). Concept learning and the problem of small disjuncts. IJCAI'89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence. pp.813-818. San Mateo, Morgan Kaufmann.

Kohavi, R. (1994). Bottom-up induction of oblivious read-once decision graphs: strengths and limitations. AAAI'94: Proceedings of the Twelfth National Conference on Artificial Intelligence. pp.613-618. Menlo Park, California, AAAI Press/MIT Press.

Michalski, R.S. and Chilausky, R.L. (1980). Knowledge acquisition by encoding expert rules versus computer induction from examples—A case study involving soyabean pathology. International Journal of Man-Machine Studies 12 63-87.

Muggleton, S. (1987). Structuring knowledge by asking questions. Bratko, I. and Lavrac, N., Ed. Progress in Machine Learning. pp.218-229. Wlmslow, UK, Sigma Press.

Murphy, P.M. and Aha, D.W. (1994). UCI Repository of machine learning databases. Department of Information and Computer Science, University of California at Irvine. ftp://ics.uci.edu.

Oliver, J.J. (1993). Decision graphs - an extension of decision trees. Proceedings of 4th International Workshop on AI and Statistics. pp.343-350.

Quinlan, J.R. (1979). Discovering rules by induction from large collections of examples. Michie, D., Ed. Expert Systems in the Micro Electronic Age. pp.168-201. Edinburgh, Edinburgh University Press.

Quinlan, J.R. (1991). Foreword. Piatetsky-Shapiro, G. and Frawley, W., Ed. Knowledge Discovery in Databases. pp.ix-xii. Cambridge, Massachusetts, MIT Press.

Quinlan, J.R., Ed. (1993). C4.5: Programs for Machine Learning. San Mateo, California, Morgan-Kaufman.

Shapiro, A.D. (1987). Structured Induction in Expert Systems. Wokingham, UK, Addison-Wesley.