An Overview of Current Research on Knowledge Compilation and Speedup Learning

Douglas Fisher
Department of Computer Science
Vanderbilt University
Nashville, TN 37235

Devika Subramanian
Department of Computer Science
Cornell University
Ithaca, NY 14853

Prasad Tadepalli
Department of Computer Science
Oregon State University
Corvallis, OR 97331

This volume contains the papers accepted for the informal workshop on Knowledge Compilation and Speedup Learning held along with the Machine Learning Conference in Aberdeen, Scotland. This workshop is a sequel to the first Knowledge Compilation workshop, which was organized by Jim Bennett, Tom Dietterich, and Jack Mostow in Otter Crest, Oregon, USA in 1986.

"Knowledge Compilation" is the problem of converting a declarative specification or a domain theory to an efficient executable program. "Speedup Learning" is the problem of improving, or speeding up, a slow problem solver. These two tasks are closely related, since a declarative specification or a domain theory can be viewed as a slow problem solver.

The approaches to knowledge compilation or speedup learning include empirical learning, explanation-based learning (EBL), and partial evaluation, among others. As the papers in this volume demonstrate, the differences between these various approaches are subtle, and partly a matter of emphasis and interpretation. Roughly speaking, "empirical learning" emphasizes learning efficient classification rules by generalizing the similarities and differences among the training examples. "Explanation-based learning" emphasizes the importance of "explanations" or problem-solving traces in identifying the relevant aspects of the training examples for generalization, and "partial evaluation" views learning as specializing or partially evaluating the domain theory, deemphasizing the role of training examples.

Speedup learning has seen a tremendous growth in recent years with the resurgence of interest in explanation-based learning (Dejong & Mooney 1986) (Mitchell, Keller & Kedar-Cabelli 1986) and chunking (Laird, Rosenbloom & Newell 1986), after the long lull that followed the early papers on robot learning in STRIPS (Fikes, Hart & Nilsson 1972). While there were some positive results in EBL (Minton 1990), the "utility problem," or the exorbitant computational cost of using the learned knowledge, proved to be a significant obstacle to further progress (Minton 1990; Tambe, Newell & Rosenbloom 1990). On the other hand, there have also emerged some interesting relationships between partial evaluation and EBL, e.g., (van Harmelen & Bundy 1988), and between EBL and empirical learning, e.g., (Yoo & Fisher 1991). The aim of this workshop was to consolidate what we learned from all these areas and define the next round of research problems and issues.

If the papers of this volume are any indication, Knowledge Compilation and Speedup Learning is a thriving subfield of machine learning. The boundaries between the various approaches to speedup learning are becoming increasingly fuzzy, demanding more rigorous and precise descriptions of methods; the techniques are getting more and more mature and sophisticated; and the evaluations are increasingly more thorough and conclusive. Summarizing the research activity in such a dynamic field is indeed a challenge. However, some tentative patterns are emerging. We clustered the papers under three headings: Empirical Speedup Learning, Tradeoffs in Knowledge Compilation, and Extensions and Evaluations of Speedup Learning Mechanisms. This division is mostly pragmatic, and almost every paper in this volume can be classified under more than one heading. Hence these divisions should not be viewed as rigid classifications, but merely as perspectives. Whenever possible, we try to identify the connections between the different pieces of work. But we are also sure that there are many more connections that we have left out.

Without further ado, we plunge in and see what the field offers.

1 Empirical Speedup Learning

As Flann's (this volume) work demonstrates, knowledge compilation can be done directly from the domain theory without any examples. In the terminology of (Dietterich 1986) this is an example of symbol level learning. However, many speedup learning systems are *empirical* in the sense that they extract and exploit the problem distribution information from the examples. The word "empirical" here should not be viewed as the opposite of "explanation-based." Since the explanation-based approaches to speedup learning exploit the distribution information as well, they are also *empirical* in the true sense of the word.

In fact, the papers in this volume amply demonstrate that the distinction between the "explanation-based" and "empirical" approaches to speedup learning is blurry and of dubious value. The papers by Fisher et al., Laird, Zelle & Mooney, Holder, Zweben & Davis, McCluskey & Porteous, and Borrazo et al., (all this volume) nicely combine the aspects of both "explanation-based" and "empirical" approaches.

The paper by Fisher, Manganaris and Yoo (this volume) is a good example of this integrated perspective. Fisher et al. eagerly embrace the connection between good categorization and efficient problem solving, which directly leads to a corresponding connection between "empirical" concept learning and "explanation-based" speedup learning. In particular, their system EXOR builds a hierarchy of categories that correspond to "explanations" at various boundaries of operationality. Problem solving then directly corresponds to classifying the problem into one of the complete explanations at the leaves of the category hierarchy. This explicit recognition of the correspondence between classification and problem solving allows them to identify some interesting relationships between EBL and empirical learning. For example, the utility problem in EBL which arises due to a large number of overly-specific explanations is seen to correspond to the overfitting problem in empirical learning, a view also echoed by Holder (this volume), and a solution is proposed based on the judicious pruning of the nodes of the category hierarchy. Similarly, controlling the search in problem solving translates to inferring the feature values which are highly predictive of the solution paths and are cost effective. Another important benefit that automatically falls out of the category hierarchy is that the rule utility is measured in a context-dependent way. This paper demonstrates that it is not only possible to integrate the explanation-based and empirical approaches to speedup learning, but also they are indeed two sides of the same coin. Another contribution of this paper is an application of a new clustering technique to fault diagnosis, where in, a "fault tree" is restructured to yield a categorization hierarchy with the minimum expected cost/benefit ratio.

Laird (this volume) shares with Fisher et al. the goal of searching for a program with a near-optimal expected cost. Laird as well as Zelle and Mooney (this volume) attack the task of optimizing logic programs. Both Laird and Zelle & Mooney advocate directly transforming the logic (Prolog) programs to more efficient versions, rather than learning separate control rules, which have a significant overhead. Laird's approach is to apply a series of correctness-preserving transformations to the input program. In this view, Explanation-Based Generalization (EBG) is just one (called "unfolding") of the many possible transformations. Interestingly, he shows that this unfolding transformation can be understood as a special case of his more general reordering transformation. His approach consists of iteratively transforming the program and then testing it on a set of problems to see if the performance has improved. While the correctness of the resulting program is ensured by the soundness of the transformations, the justification for the expected speedup is based on purely *empirical* grounds: the output program is expected to be near-optimal for the given problem distribution if it is near-optimal for a big enough sample of training problems drawn from that distribution.

Zelle and Mooney (this volume) explicitly combine EBG with an inductive learning program called FOIL (Quinlan 1991), which learns relational knowledge from examples. The precondition literals extracted by EBG from the problem solving traces of a logic program are used to identify potential relevant literals for the induction program. The induction program chooses a minimal subset of these literals necessary to distinguish the positive and negative examples of when a Prolog clause should be applied. By filtering the number of potential relevant literals using EBG's proofs, the number of training examples needed for convergence in FOIL is drastically reduced. Similarly, the use of FOIL reduces the number of literals that would otherwise occur in the precondition proofs of EBG, thus avoiding the utility problem and the generalization-to-N problem. Their DOLPHIN system is a culmination of a number of systems that successfully integrated explanation-based and empirical learning, most notably the IOE system of (Flann & Dietterich 1989) and the AxA-EBL of (Cohen 1990).

Holder (this volume) also explicitly calls for a unification of explanation-based and empirical learning systems. In both paradigms the *utility* (Minton 1990) of learned knowledge stems from a tradeoff between coverage and performance improvement, which is typically measured by accuracy or time efficiency in empirical and explanation-based approaches, respectively. His system, MPAC, which can be coupled with a variety of learning systems, traces out learning curves for the component learning system over a number of trials. A pervasive pattern in these curves is that after initial improvement, performance degrades due to the acquisition of overspecialized knowledge – an overfitting of learned rules to the available data. MPAC fits the curve with a parabola, estimates the point of peak performance (i.e., the parabolic vertex), thus separating useful learned knowledge (i.e., knowledge learned to 'left' of the vertex) from detrimental knowledge. Holder also suggests a formal approach to estimating performance peaks, though there are probably others such as Bayesian approaches that estimate the 'most probable' model for observed data (Cheeseman et al. 1988) or principles of minimum description length representations. As we will see, Holder's discussion seems to imply that like Flann (this volume), the acquisition of 'desirable' knowledge is roughly ordered by decreasing coverage.

Another piece of work that blurs the distinction between "explanation-based" and "empirical" approaches is by Zweben and Davis (this volume). Zweben and Davis employ EBL in an unlikely application – iterative repair scheduling. The overall task here is to construct correct schedules for the NASA space shuttle by iteratively repairing the constraint violations in a complete schedule. Their version of EBL called *Plausible EBL* (PEBL) uses multiple examples to confirm conjectured explanations. PEBL was used to select between two repair strategies – a less informed, "depth 0 strategy," and a more informed, "variable depth strategy." The explanations obtained using the depth 0 strategy are only "plausible," because the exact explanations require an expensive search. Hence these rules/explanations are *empirically* evaluated, yielding confidence values. When the confidence in a set of plausible rules relevant to a problem is high, then the depth 0 strategy is used. In other cases, the more expensive, variable depth strategy is employed. By being able to choose between the two strategies in a problem-dependent manner, the learning system is able to outperform the non-learning system whether it uses the depth 0 or the variable depth strategy. Viewed from the perspective of empirical learning, PEBL is a generator of inductive hypotheses, which are verified by the training sample.

McCluskey and Porteous (this volume) combine three different learning algorithms in their FM system: a goal reformulation component similar to the GENCOM algorithms of (Irani & Cheng 1989), an abstraction construction component similar to that of (Knoblock 1990), and a control rule learning component similar to EBG, but which learns approximate heuristic control rules by dropping some of the precondition literals. While the goal reformulation and the abstraction components do not have an empirical aspect to them, the control rule learning component does, and the main contribution of this paper is a demonstration that these three algorithms exploit different sources of power in at least some domains, so that combining all the three would result in a better performance than using any one of them. This result seems to be consistent with that of (Knoblock et al., 1991). Interestingly, the abstraction mechanism alone outperformed the combination in a two agent STRIPS world! The authors attribute this to the overhead of doing goal reformulation, which yielded only modest benefits in terms of CPU seconds saved, which is their performance measure. This points to the difficulty of finding a robust performance measure which is uniformly applicable across widely different learning techniques.

Borrazo, Caraça-Valente and Pazos (this volume) also describe a general learning architecture based on learning of specific control rules by remembering good and bad operator choices in state space search. They have plans to add an empirical learning component on top of this to generalize the control rules. One of the advantages of this approach is that it does not require any domain knowledge other than the operator definitions, which can be opaque, as suggested in (Porter & Kibler 1986) and Tadepalli & Isukapalli, this volume. Their current approach seems to share some of the aspects of Q-learning in that each new example propagates the values of the states (or, equivalently, the goodness of operator choices) backward from the goal (Watkins 1989). Their system has been applied to some two-person game domains, and, in contrast with Flann's "abstract reverse enumeration" (this volume), uses only the concrete state space, and learns from example games played against an opponent.

2 Extensions and Evaluations of Speedup Learning Mechanisms

The study of speedup learning methods both of the EBL and search-based macro learning variety has significantly matured. Evaluations of speedup methods that document interactions between the generation and use of the structures produced by these mechanisms is the topic of the papers by Iba, Markovitch et al., and Etzioni et al. Two extensions to speedup learners are explored in here: the integration of rule composition learning with abstraction by Bergmann, and scaling up of EBL to a complex knowledge assimilation problem by Levi et. al.

Iba (this volume) presents an experimental study of the interaction between policies for the generation and use of macros in search problems: the sliding tile puzzle and peg solitaire (generalizations of the 8-puzzle and hi-Q respectively). Two macro-generation policies are used: one is a small modification of the peak-to-peak heuristic in (Iba 1989), the other involves hand selection of macros generated by the previous method. Two policies for macro use are investigated: *always* and *impasse-only*. In the *always* mode, macros are available at all times and thus they increase branching factor. In the *impasse-only* mode, search proceeds using the primitive operators until a local peak in the evaluation function is reached: macros are employed to jump over local peaks. Iba provides evidence that hand-selected macros and the always use mode work well in the peg solitaire domain. In the sliding tile domain results were inconclusive – there is a tradeoff involving solution lengths and the macro use and generation policies. This suggests that domain properties and problem distributions need to be linked to find qualitative laws for macro generation and use in search-based solvers.

In another experimental study, Markovitch and Rosdeutscher (this volume) investigate the same interaction between macro generation and use schemes in the context of the weighted A* algorithm. The search strategy used during learning controls the macros that are generated, the search strategy used during problem solving determines how the macros are used. All subpaths of a successful solution path are stored during the learning phase. These subpaths form the macros that are used during subsequent problem solving. The interesting result here is that macros degrade the performance of problem solvers like A* that guarantee optimal solutions. The reason is that the redundancy introduced by the macros forces the algorithm to search a greater number of paths. The experimental results in a simplified grid world suggest that macros should be *learned* in the context of a search algorithm that is optimal (A*) or close to optimal, and *used* by best-first searchers as opposed to the searchers using A*.

Etzioni & Minton[1] consider the generation problem in the context of a specific use policy: which control rules to extract from a problem-solving trace (logical proof) in the Prodigy system. These rules are used by a meta-level solver that matches them in each decision cycle. EBL methods considered thus far in the context of Prodigy, use a specific example to generalize the conditions under which an action or action sequence can be taken. Therefore, they miss common features across examples. To overcome this limitation, Etzion and Minton propose the construction of minimal sufficient conditions over the partial-order $\leq$ defined over well-formed formulas as follows: $s_1 \leq s_2 \equiv s_2 \models s_1 \;\wedge\; \texttt{simpler}(s_2, s_1)$ where `simpler` is a syntactic ordering between formulas that captures the cost of matching. Common sub-expression elimination, abstractions by dropping conditions are proposed as heuristic mechanisms for learning minimal descriptions which produce efficiency improvements at the decision-making level regardless of the distribution of examples. The approach advocated in this paper is a domain-independent meta-theoretic criterion for the selection of appropriate control rules in Prodigy.

To produce more powerful speedup methods, we have to extend the range of compilations of domain theories that are generated. Traditional EBL methods learn compositions of rule sequences that short-circuit chains of reasoning. Traditional methods of abstraction like Abstrips, search in reduced spaces by constructing equivalence classes of base-level states. Bergmann (this volume) integrates EBL and abstraction methods (Knoblock et. al, 1991) by devising a scheme that learns abstract state descriptions as well as compositions of rule sequences using a generic theory of abstraction as a domain theory. This method properly generalizes the methods of (Knoblock 1990, (Anderson & Farley 1988), (Unruh & Rosenbloom 1988).

The paper by Levi et. al. (this volume) is an interesting application of EBL to a knowledge assimilation task – macros produced by standard declarative EBL methods in the learning component of a complex, hybrid system are integrated with an existing procedural representation employed by the performance element of the system. The task considered is developing and maintaining knowledge-bases in the Pilot's Associate program. The Pilot's Associate is a coordinated suite of 6 expert systems that aid the pilot of an advanced tactical fighter. The representations employed by these expert systems (which include a tactical planner) are highly procedural and use vocabulary far-removed from that of the experts. The challenge here is to learn new strategies from simulator traces generated by expert flyers, and to integrate them seamlessly into the various expert systems without human intervention. Levi et. al. provide temporal extensions involving interval aggregation needed to the basic EBL algorithm (Dejong & Mooney 1986) to solve the problem of learning macros from a time-stamped trace of an expert flyer's actions and a domain theory. The macro is then redescribed in the vocabulary of the tactical planner using a set of translation rules designed by the authors in conjunction with the developers of the expert system. In contrast with most knowledge compilation work which emphasizes improving problem solving efficiency, this research focuses on the *assimilation* of macros generated by a declarative EBL process, into a complex procedural problem solver. The results presented are very encouraging and demonstrate the scalability of EBL methods to real-world knowledge compilation tasks.

3 Tradeoffs in Knowledge Compilation

Recently, some researchers have suggested that knowledge compilation insists that the consequences of a domain model or theory be fully enumerated, and that simulation/search with the model/theory be displaced entirely by a kind of 'lookup' of condition-action pairs (Ginsberg 1989). For example, a learning system by Pearce (1988) 'compiles' model-based fault simulations into an exhaustive rulebase using AQR (Michalski & Larson 1983) for purposes of diagnosis. However, this strategy is a 'strawman' that most machine learning researchers do not adhere to. In fact, the 'speedup' aspect of this Workshop makes their bias explicit: if compilation does not lead to speedup, then compilation is being employed improperly, too aggressively, and/or under inappropriate circumstances.

[1]This volume contains only the abstract. Please see Proceedings of the Machine Learning Conference, 1992 for the full paper.

Flann (this volume) demonstrates the qualifications on knowledge compilation nicely. In particular, he begins by considering the merits of a full enumeration, 'strawman' procedure, but notes the space and time disadvantages underlying it. In the context of counterplanning (e.g., game playing), Flann shows that abstraction can significantly compress the size of the rulebase that results from compilation. In fairness, abstraction also naturally emerges from the use of AQR in Pearce's application. However, Flann goes beyond this, and points out that abstractions differ in their coverage, and thus the degree to which they facilitate speedup. Faced with very large domain theories in which full enumeration is intractable, Flann's system heuristically enumerates rules from greatest coverage to least coverage. In a chess endgame domain, he shows that as much as 50% of the search space is covered by generated rules in a fraction of the time required by the compiler to reach full coverage. The implications are clear: when faced with time constraints, the compiler can quickly achieve significant coverage, thus facilitating substantive speedup in subsequent problem solving, even though the compiled rules alone are not sufficient to insure a complete problem solver.

Like Flann, Barley (this volume) is also concerned with a time/coverage tradeoff, though his interpretation of the two factors participating in this tradeoff have different connotations. In particular, Barley argues that the traditional correctness and completeness preserving approaches to speedup in planners (e.g., macro learning) is inherently limited. In general, robust speedup must also rely on knowledge level learning, in which the planner explicitly reasons about and modifies its planning biases. In general, but with caveats, Barley argues that more restrictive biases limit the search space explored by a planner, thus promoting 'speedup', but at the cost of completeness. Several systems, including the author's BACALL, relax a planner's bias in response to planning failures, thus increasing the planner's coverage – an ability that can be interpreted as knowledge level learning depending on how one delimits the performance system. On the surface, this work might seem very different from Flann's, but both are intimately concerned with a tradeoff that is at the heart of speedup learning. Notably, Flann's concern with the time of compilation prior to problem solving is matched by Barley's concern with the *interpretation* time needed for problem solving by interactively relaxing bias. In addition, incremental bias adjustment serves to increase the coverage, and it is the difference in coverage between successive adjustments that apparently corresponds to Flann's enumeration of compiled rules that incrementally increases their coverage.

On the surface, Koedinger (this volume) is not directly concerned with the tradeoffs outlined above. Instead, he challenges speedup researchers to look at models of human problem solving for insights into learning. In particular, his earlier experimental findings and those of others suggest that experts in certain domains are largely forward reasoners, and are guided little by the goal. Koedinger's DC model accounts for this behavior in geometry by exploiting loosely-coupled problem-solving chunks that can be combined at problem-solving time to make a wide variety of inferences, some of which are bound to satisfy the subgoals for any given problem. These chunks are based on perceptual regularities that are an integral part of geometry problem solving, and may relate to the geometric representations promoted by Flann, though we did not highlight this aspect of Flann's contribution. Moreover, Koedinger gives a nice analysis of how DC's perceptual schemas might map to the kind of rule-based representations often employed by speedup learning researchers. Comparisons between expert protocols and an ACT*-based compilation mechanism that combines rules that occur successively a sufficient number of times, reveals that the approach does not converge on the same set of schemas that are apparently present in human experts. Koedinger concludes by speculating on a number of possible biases that might facilitate such a convergence.

Interestingly, one aspect of Tadepalli and Isukapalli's (this volume) research provides a strategy that might promote convergence on rule-based analogs to Koedinger's DC-like schemas. They argue that the true source of power in macro-learning systems is deciding upon the 'best' grain size of macros, which can then be flexibly combined during planning. In particular, their COMPOSER system derives macros that are composed of strongly-coupled operators in partially-ordered plans. Coupling takes the form of clobbering relations of the type employed in nonlinear planners such as TWEAK (Chapman, 1987). It is this focus on partial (versus linear) ordering and on coupling (versus frequency) criteria that could speak to the learning requirements present in Koedinger's application. In focusing on appropriate grain sizes for macro operators, Tadepalli and Isukapalli also link their (and Koedinger's) work to more general issues of coverage: minimal-sized macros will tend to have greater applicability, and strong within-macro coupling (and correspondingly weak between-macro coupling) will better insure that those portions of the search space that are not covered by macros will nonetheless be amenable to a strongly-biased, efficient search. Should problem-solving fail then the authors point out that the bias can be relaxed, as suggested by Barley (this volume), by appealing to a less restricted planner. These general observations also underlie the work of researchers like Zelle & Mooney (this volume), and (Ruby & Kibler 1989).

In closing this summary of papers, we should note that a second contribution of Tadepalli and Isukapalli is their promotion of learning with opaque operators or from a simulator, which cannot be fully inspected as can declarative representations. In forwarding such a proposal, they explicitly point out that opaque representations reduce the ability of the analyst to engineer a domain in a way that highlights the particular advantages of a speedup approach, and hides its corresponding weaknesses. It would be interesting to see if their techniques for learning from simulators would scale to such real world domains as Pilot's Associate, which are being explored by Levi et al. (this volume).

4 Concluding Remarks

Our survey indicates the breadth of research underway in knowledge compilation and speedup learning, but there are a healthy dose of common issues that we expect this workshop to bring to the foreground. In addition, though we have not emphasized it here, we believe that speedup approaches have matured to a point that demands sophisticated evaluation methods that are beginning to take root in empirical learning paradigms. For example, we believe that we should be able to reasonably conclude that a method improves performance to the *extent possible* or to some approximation of this ideal; *absolute* measures of speedup are of dubious validity, invite domain engineering, and should be displaced by evaluations of the *proportion* of theoretical speedup that is realized. In some cases, there may be no speedup opportunities in a well engineered domain theory at all. Rather, speedup learning is a tool for domain engineering that should be used appropriately (Davis 1989; Keller 1990). We hope that the Workshop brings out appropriate contexts for knowledge compilation and speedup learning, and other issues of evaluation as well.

5 Acknowledgments

We thank Pat Riddle for indirectly prompting us to write this overview, sharing her overview with us, and proof reading ours. We thank Ramana Isukapalli for proof reading this and helping us put the proceedings together.

References

[Anderson, J. S. & Farley, A. M., (1988)] Plan Abstraction Based on Operator Generalization. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann.

[Chapman, D. (1987)] Planning for Conjunctive Goals. *Artificial Intelligence*, 32:333-337.

[Cheeseman, P. Kelly, J. Self, M. et al. (1988)] Autoclass: A Bayesian Classification System. In *Proceedings of the Fifth International Conference on Machine Learning*, San Mateo, CA: Morgan Kaufmann.

[Cohen, W. (1990)] Learning Approximate Control Rules of High Utility. In *Proceedings of the Seventh International Conference on Machine Learning*, San Mateo, CA: Morgan Kaufmann.

[Davis, R. (1989)] Form and Content in Model-based Reasoning. In *Proceedings of the 1989 Workshop on Model-Based Reasoning.*

[Dejong, G. & Mooney, R. (1986)] Explanation-Based Learning: An Alternative View. *Machine Learning*, 2, 145-176.

[Dietterich, T. G. (1986)] Learning at the Knowledge Level. *Machine Learning*, 1, 287-316.

[Fikes, R. E., Hart, P.E., & Nilsson, N. J. (1972)] Learning and Executing Generalized Robot Plans. *Artificial Intelligence*, 3, 251-288.

[Flann, N. S. & Dietterich, T. G. (1989)] A Study of Explanation-Based Learning for Inductive Learning. *Machine Learning*, 4, 187-226.

[Ginsberg, M. (1989)] Universal Planning: An (Almost) Universally Bad Idea. *AI Magazine*, 4(10).

[Iba, G. (1989)] A Heuristic Approach to the Discovery of Macro-operators. *Machine Learning*, 3(4), 285-317.

[Irani, K. B. & Cheng, J. (1989)] Ordering Problem Subgoals. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann.

[Keller, R. M. (1990)] In Defense of Compilation. In *Proceedings of the Second AAAI Workshop on Model Based Reasoning* (pp. 22-31).

[Knoblock, C. A., (1990)] Learning Abstraction Hierarchies for Problem Solving. In *Eighth National Conference on Artificial Intelligence*, Menlo Park, CA: AAAI Press.

[Knoblock, C. A., Minton, S., & Etzioni, O. (1991)] Integrating Abstraction and EBL in PRODIGY. In *Ninth National Conference on Artificial Intelligence*, Menlo Park, CA: AAAI Press.

[Laird, J. E., Rosenbloom, P. S., & Newell, A. (1986)] Chunking in Soar: The Anatomy of a General Learning Mechanism. *Machine Learning*, 1, 11-46.

[Michalski, R. S. & Larson, J. B. (1983)] Incremental Generation of VL_1 Hypotheses: The Underlying Methodology and the Description of Program AQ11. Technical Report ISG 83-5, The University of Illinois at Urbana-Champaign, Department of Computer Science, Urbana, 1983.

[Minton, S. (1990)] Quantitative Results Concerning the Utility of Explanation-Based Learning. *Artificial Intelligence*, 42, 363-392.

[Mitchell, T. M., Keller, R., & Kedar, S. (1986)] Explanation-Based Generalization: A Unifying View. *Machine Learning*, 1, 47-80.

[Pearce, D. A. (1988)] The Induction of Fault Diagnosis Systems from Qualitative Models In *Seventh National Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann.

[Porter, B. W., & Kibler, D. F. (1986)] Experimental Goal Regression: A Method for Learning Problem-Solving. *Machine Learning*, 3(1).

[Quinlan, J. R. (1991)] Learning Logical Definitions from Relations. *Machine Learning*, 5, 239-266.

[Ruby, D. & Kibler, D. (1989)] Learning Subgoal Sequences for Planning. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann.

[Tambe, M., Newell, A. & Rosenbloom, P. S. (1990)] The Problem of Expensive Chunks and its Solution by Restricting Expressiveness. *Machine Learning*, 5 (3), 299-348.

[Unruh, A. & Rosenbloom, P. S. (1989)] Abstraction in Problem Solving and Learning. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann.

[van Harmelen, F. & Bundy, A. (1988)] Explanation-Based Generalization = Partial Evaluation. *Artificial Intelligence*, 36, 401-412.

[Watkins, C. (1989)] *Learning from Delayed Rewards* Ph.D. thesis, Cambridge University, 1989.

[Yoo, J. & Fisher, D. (1991)] Concept Formation over Explanations and Problem Solving Experience. In *Proceedings of IJCAI-91*, San Mateo, CA: Morgan Kaufmann, 1991.