

## A Transformational Analysis of the EBL Utility Problem

Jihie Kim and Paul S. Rosenbloom

Information Sciences Institute and Computer Science Department  
University of Southern California  
4676 Admiralty Way  
Marina del Rey, CA 90292, U.S.A.  
jihie@isi.edu, rosenbloom@isi.edu

Efficiency is a major concern for all problem solving systems. One way of achieving efficiency is the application of learning techniques to speed up problem solving. Accordingly, there has been considerable amount of research on applying explanation-based learning (EBL)(Mitchell, Keller, & Kedar-Cabelli 1986) techniques to problem solving. However, EBL is known to suffer from the *utility problem*, where the cost of using the learned knowledge overwhelms its benefit. We show that how the cost increase of a learned rule in an EBL system can be analyzed by characterizing the learning process as a sequence of transformations from a problem solving episode to a learned rule. The analysis of how the cost changes through the transformations can be a useful tool for revealing the sources of cost increase in the learning system. We focus on the Soar problem solving system which uses a variant of EBL called *chunking*(Rosenbloom *et al.* 1991). The chunking process has been decomposed into a sequence of transformations from the problem solving to a chunk (learned rule). By analyzing these transformations, we have identified a set of sources which can make the output chunk expensive. The set of sources and the proposed solutions are :

1. *Removing search control*: Ignoring search-control rules which constrained the problem solving can increase the cost. For example, PRODIGY/EBL (Minton 1993) and Soar ignore a large part of the search-control rules in learning to increase the generality of the learned rules. The consequence of this omission is that the learned rules are not constrained by the path actually taken in the problem space, and thus can perform an exponential amount of search even when the original problem-space search was highly directed (by the control rules). By incorporating search control in the explanation structure, the match process for the learned rule can focus on the path that was actually followed (Kim & Rosenbloom 1993).
2. *Eliminating intermediate rule firing (Unifying)*: In Soar, working memory is a set. Whenever two different instantiations create the equivalent working memory elements, they are merged into one. Eliminating this process in learning, and keeping equivalent set of partial instantiations separately can increase the cost. By preprocessing the partial instantiations, and merging the equivalent instantiations into one, this cost increase can be avoided.

This could potentially be done either by grouping instantiations that generate the same working memory elements or by selecting one of them as a representative.

3. *Totally ordering the conditions (Linearize)*: Simplifying the structure of learned rule without considering the problem solving structure can increase the cost. By making the learning mechanism sensitive to the problem-solving structure, — i.e., by reflecting such structure in the match of the learned rule — we can avoid this source of expensiveness. The key thing that this requires is an efficient generalization for the match algorithm to interpret the non-linear structure (Kim & Rosenbloom 1996).

Based on the above analysis and the proposed potential solutions to the sources of expensiveness, we are currently working towards the specification and implementation of a variant of chunking which avoids any of these sources of extra cost. If it works, the cost of using a chunk should always be bounded by the cost of the corresponding problem solving. A similar transformational analysis should also be possible for any problem solving systems which use EBL techniques. As with the analysis of chunking, this analysis should identify sources of expensiveness in EBL, and help guide the design of safer EBL mechanisms(Kim & Rosenbloom 1995).

### References

- Kim, J., and Rosenbloom, P. S. 1993. Constraining learning with search control. In *Proc. 10th Int'l Conf. on Machine Learning*, 174–181.
- Kim, J., and Rosenbloom, P. S. 1995. Transformation analyses of learning in Soar. Technical Report ISI/RR-95-4221, USC-ISI.
- Kim, J., and Rosenbloom, P. S. 1996. Learning efficient rules by maintaining the explanation structure. In *Proc. 13th Nat'l Conf. on Artificial Intelligence (to appear)*.
- Minton, S. 1993. Personal communication.
- Mitchell, T. M.; Keller, R. M.; and Kedar-Cabelli, S. T. 1986. Explanation-based generalization – a unifying view. *Machine Learning* 1(1):47–80.
- Rosenbloom, P. S.; Laird, J. E.; Newell, A.; and McCarl, R. 1991. A preliminary analysis of the soar architecture as a basis for general intelligence. *Artificial Intelligence* 47(1-3):289–325.