IMEX: Overcoming Intractability in Explanation Based Learning*

Michael S. Braverman Stuart J. Russell
573 Evans Hall
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720

Abstract

Compiled knowledge, which allows macro inference steps through an explanation space, can enable explanation-based learning (EBL) systems to reason efficiently in complex domains. Without this knowledge, the explanation of goal concepts is not generally feasible; moreover, the problem of finding the most general operational concept definition is intractable. Unfortunately, the use of compiled knowledge leads to explanations which yield overly specific concept definitions. These concept definitions may be overly specific in one of two ways: either a similar concept definition with one or more constants changed to variables is operational, or a concept definition which is more general, according to the implication rules of the domain theory, is operational. This paper introduces a method (IMEX) for modifying, in a directed manner, the explanation structures of goal concepts that have been derived using compiled knowledge. In this way, more general operational concept definitions may be obtained.

1. Introduction

The methods of explanation based learning (EBL) [Dejong & Mooney, 1986] and explanation based generalization (EBG) [Mitchell, Keller, & Kedar-Cabelli, 1986] involve two conceptual phases: explanation and generalization. Until recently, little consideration has been given to the dependencies of the generalization phase upon the explanation phase or to the difficulties of forming the explanation itself. These two factors are strongly influenced by the form and content of the domain theory being used by the explanation based method.

Various researchers have noted that the goal of the explanation based methods is not only to generalize, but also to produce generalizations that are easy to apply in future situations. This ease of application is captured by the notion of operationality as defined by Mitchell, *et al.* [1986] and extended by Dejong and Mooney [1986] and Keller [1987]. We adopt the method for evaluating operationality suggested by Hirsh [1987,1988] and Mostow [1987]; namely, the operationality of a given concept definition is determined by supplied rules which allow deliberate meta-reasoning about the knowledge in the domain theory. We have the compound goal of finding explanation structures that yield concept definitions that are not only operational, but also maximally general.

With *intractable* domain theories [Mitchell *et al.*, 1986], however, it may be difficult to form even a single explanation, let alone find the best one for generalization purposes. One approach for dealing with this problem is to admit approximations to the domain theory that allow quicker explanations at the expense of accuracy [Ellman, 1988; Bennett, 1987]. Alternatively, and without loss of accuracy, the problem of finding explanations in complex domains may be made more tractable if the explanation module is given knowledge that allows macro inference steps in the explanation space; herein, we refer to this type of knowledge as compiled knowledge. The use of compiled knowledge to achieve efficiency is, of course, not new; Scripts [Cullingford, 1978] for story understanding, MACROPs [Fikes, Hart, & Nilsson, 1972] for robot planning, and Chunks [Laird, Rosenbloom, & Newell, 1986] for general problem solving are three notable examples. Korf [1987] has shown that the use of macro-operators in abstraction hierarchies can reduce the complexity of problem solving from exponential to linear.

Indeed, the very point of EBL is to create compiled knowledge in order that the performance element of the system may operate more efficiently in the future. Unfortunately, as will be shown in Sections 2 and 3, the use of compiled knowledge leads to explanations that give less general concept definitions than would otherwise be obtained without its use. Given a domain theory consisting of logical axioms, a concept $q(\vec{x})$ is at least as general as a concept $p(\vec{x})$ if it can be shown that $p(\vec{x}) \rightarrow q(\vec{x})$. From this it follows that a concept $r(\vec{x})$ over a vector of uninstantiated variables $\vec{x}$ is more general than the same concept with one or more of the variables instantiated. The straight forward use of compiled knowledge leads to overly specific concept definitions in two ways: concept definitions are produced in terms of $p(\vec{x})$, even though $q(\vec{x})$ is operational and $p(\vec{x}) \rightarrow q(\vec{x})$; and concept definitions are produced in terms of $r(\vec{y})$, with the elements of $\vec{y}$ unnecessarily or overly instantiated. There is an inherent conflict between being able to find any explanation at all (using compiled knowledge) and obtaining desirable generalizations (not using compiled knowledge).

This paper introduces a method, called IMEX, to *Incrementally Modify* a given *EXplanation* to make it better meet

*We would like to acknowledge assistance from members of BAIR, the Berkeley Artificial Intelligence Research Project, under the direction of Robert Wilensky. The first author is an AT&T Bell Laboratories Scholar. This research is sponsored in part by that scholarship and by the Defense Advanced Research Projects Agency (DoD), Arpa Order No. 4871, monitored by Space and Naval Warfare Systems Command under Contract N00039-84-C-0089. The research is also supported by grants to the second author from the University of California MICRO program and Lockheed AI Center.

the criteria of operationality and generality. We assume that the domain theory used to construct the given explanation contains compiled knowledge. IMEX then uses the operationality criteria to focus on those parts of the explanation that should be changed in order to obtain a more useful explanation structure. Thus, the operationality criterion is used to *motivate* explanation modifications, in contrast to other approaches that generate all possible explanations and then use the operationality criterion as a *filter*.

2. Implication Rules and Generality

2.1. The Boundary of Operationality

For any generalized *explanation structure* [Mitchell *et al.*, 1986], if we remove one or more rules from the bottom of the structure, we obtain a new structure whose conjunct of leaf nodes yields a potentially more, and certainly not less, general concept definition than would the original structure. The *boundary of operationality* [Braverman and Russell, 1988] of an explanation structure is the highest line that can be drawn through the structure such that if the rules supporting the nodes immediately below the boundary were eliminated, the resulting structure would yield an operational concept definition. If the boundary line were moved any higher, then the concept definition of the new structure would be non-operational. Thus, the boundary locates the concept definition which, according to the implication rules of the domain theory, is the most general operational concept immediately derivable from the explanation structure.

Consider the following example which is a modified version of the Safe-To-Stack example from [Mitchell *et al.*, 1986]. Although the domain is not particularly complex, imagine that the domain contains many more axioms, making it infeasible to try all possible proofs. The domain theory contains a manufacturing constraint (rule 7) on rectangular, solid objects made of lucite. Assume that we often refer to the volume and weight of these objects during problem solving, and this has led to the creation of the two compiled pieces of knowledge in rules 9 and 10. The domain theory is as follows (where $Times(x,y,z)$ and $Less(a,b)$ are procedurally defined to be true when $z = x \times y$ and $a < b$, respectively):

$$Not(Fragile(y)) \rightarrow Safe\text{-}To\text{-}Stack(x,y) \quad (1)$$

$$Lighter(x,y) \rightarrow Safe\text{-}To\text{-}Stack(x,y) \quad (2)$$

$$Volume(p,v) \wedge Density(p,d) \wedge Times(v,d,w) \rightarrow Weight(p,w) \quad (3)$$

$$Weight(p_1,w_1) \wedge Weight(p_2,w_2) \wedge Less(w_1,w_2) \rightarrow Lighter(p_1,p_2) \quad (4)$$

$$Spec\text{-}Grav(lucite,2) \quad (5)$$

$$Madeof(p,x) \wedge Spec\text{-}Grav(x,s) \rightarrow Density(p,s) \quad (6)$$

$$Isa(p,rect\text{-}solid) \wedge Madeof(p,lucite) \wedge Length(p,l) \wedge Width(p,w) \wedge Height(p,h) \wedge Times(l,w,area) \rightarrow Times(area,h,5) \quad (7)$$

$$Isa(p,rect\text{-}solid) \wedge Length(p,l) \wedge Width(p,w) \wedge Height(p,h) \wedge Times(l,w,a) \wedge Times(a,h,v) \rightarrow Volume(p,v) \quad (8)$$

$$Isa(p,rect\text{-}solid) \wedge Madeof(p,lucite) \rightarrow Volume(p,5) \quad (9)$$

$$Isa(p,rect\text{-}solid) \wedge Madeof(p,lucite) \rightarrow Weight(p,10) \quad (10)$$

Rule 9 is a compilation of rules 7 and 8 along with the fact (not listed above) that all rectangular solid objects have some height, width, and length. Rule 10 follows from rules 3, 5, 6 and 9. We also have the following knowledge about the objects Obj1 and Obj2 (the training instance):

$Isa(Obj1,Box)$ $Isa(Obj2,rect\text{-}solid)$
$Color(Obj1,Red)$ $Color(Obj2,Clear)$
$Madeof(Obj1,wood)$ $Madeof(Obj2,lucite)$
$Spec\text{-}Grav(wood,0.1)$ $On(Obj1,Obj2)$
$Volume(Obj1,1)$

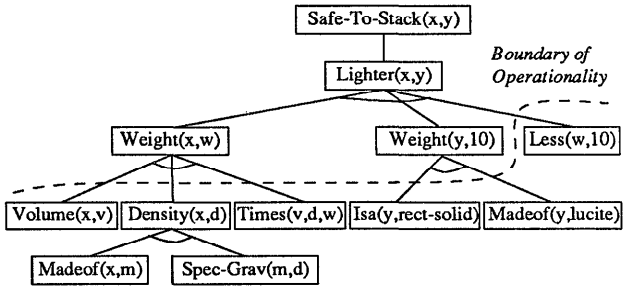

Figure 1: Generalized Explanation Structure of Safe-To-Stack(Obj1,Obj2)

Given the goal of proving *Safe-To-Stack(Obj1,Obj2)* we might derive a proof tree whose explanation structure is as shown in Figure 1. Here we assume that all predicates in the domain theory are unconditionally operational except for *Fragile*, *Lighter*, *Safe-To-Stack*, and *Weight*. Note the position of boundary of operationality. Even though the concept $Madeof(x,m) \wedge Spec\text{-}Grav(m,d)$ is operational, $Density(x,d)$ is also operational and, according to rule 6, more general. Hence, the boundary is positioned above the $Density(x,d)$ node rather than below it.

2.2. Using Compiled Knowledge: Problem One

The use of compiled knowledge to form an explanation structure can hide concept definitions which are more general, according to the implication rules of the domain theory, than the conjunct of the nodes below the structure's boundary of operationality. Given the explanation structure in Figure 1 and its associated boundary of operationality we derive the operational but not so general rule:

$$Volume(x,v) \wedge Density(x,d) \wedge Times(v,d,w) \wedge Isa(y,rect\text{-}solid) \wedge Madeof(y,lucite) \wedge Less(w,10) \rightarrow Safe\text{-}To\text{-}Stack(x,y)$$

This rule is overly specific because the compiled domain rule 10 was used in forming the explanation structure.

In order to obtain a more general concept definition, we might consider taking the explanation structure of Figure 1, removing the conjunction $Madeof(x,m) \wedge Spec\text{-}Grav(m,d)$ from the leaves (which add nothing to the concept definition), and *expanding* the compiled rule for *Weight(y,10)*; this would yield the explanation structure in Figure 2. By *expanding* a rule, we mean that the rule should be replaced, if possible, by a chain of inference steps that justify the rule. The expansion of the compiled rule reveals a new, previously hidden, concept definition that is more general according to the implication rules of the domain theory.

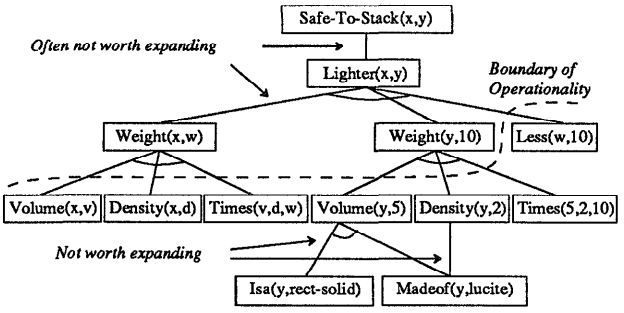

Figure 2: The Result of Expanding an Inference Step

Thus, we are motivated to remove, from the structure in Figure 2, the rules whose antecedents are the nodes *Isa(y,rect-solid)* and *Madeof(y,lucite)*. In so doing, we not only eliminate the nodes from the explanation structure, but also retract the constraints on variable values resulting from unifications of the structure with the, soon to be, removed rules. The remaining explanation structure would yield the very general (and operational) rule:

$$Volume(x_1,v_1) \wedge Density(x_1,d_1) \wedge Times(v_1,d_1,w_1) \wedge Volume(x_2,v_2) \wedge Density(x_2,d_2) \wedge Times(v_2,d_2,w_2) \wedge Less(w_1,w_2) \rightarrow Safe\text{-}To\text{-}Stack(x_1,x_2)$$

Note that expanding out some inference steps, such as those labeled *not worth expanding* above, will have no effect on the generality of the concept definition finally obtained so far as implication generality is concerned. The IMEX Implication algorithm given below is designed to effect just those changes to the explanation structure which lead to more general, operational concept definitions according to the implication rules of the domain theory.

2.3. The IMEX Implication Algorithm

Given a goal concept G to prove, the IMEX Implication algorithm may be stated as follows:

(1) Using the domain theory with all its compiled knowledge, find a proof of the goal concept G. Let E denote the explanation structure formed and compute, for E, the boundary of operationality.

(2) Take the explanation structure E and locate a rule R in the structure that straddles the boundary of operationality; i.e. all of its antecedents are directly below the line and the consequent is above the line. If no such rule can be found, then go to step (4).

(3) Try to expand the rule R; in other words, attempt to show that the consequent of R follows from its antecedents without using R itself. If this is not possible, then go to step (2) and search for another rule that straddles the boundary. If an expansion does exist, then splice it into the explanation structure E, compute the new operational boundary, and go back to step (2) with the modified E structure.

(4) Retract all rules involving nodes from E that only support other nodes below the boundary of operationality. The resulting explanation structure is the one that is used to form the general goal concept definition.

The correctness and efficiency of the algorithm are explained as follows: After step (4) only the nodes directly below the boundary of operationality will have any effect on the generality of the concept definition. Hence, in order to achieve the most general concept definition, IMEX should attempt to make the conjunction of the nodes below the boundary as general as possible. Clearly these nodes will not become more general by trying to reprove their justifications. Thus, the potentially many different expansions of inference steps of the sort indicated as *not worth expanding* in Figure 2 do not affect the generality of the final concept definition. If the operationality theory dictates that a concept definition's operationality decreases with its generality, then expanding rules whose antecedents are nodes above the current boundary will have no effect on the new boundary calculated in step (3); this follows since any new nodes that might be revealed would be part of a concept definition which is at least as, if not more, general (and, hence, less operational) than a concept definition which has already been declared non-operational by the current boundary of operationality. Therefore, the only parts of the proof definitely worth examining are those that straddle the current boundary. Step (2) checks exactly those rules. For each rule expansion, the operational boundary either stays stationary or moves up relative to the nodes originally below the boundary; the concept definition generality is monotonically non-decreasing with each IMEX iteration. By attempting to reprove those, and only those, subparts of the proof that have a definite potential of leading to a more general concept definition, the incremental algorithm drastically reduces the search space for a general explanation structure.

3. Variable Instantiation and Generality

3.1. Using Compiled Knowledge: Problem Two

The use of compiled knowledge to form an explanation structure can result in concept definitions which have unnecessarily or overly instantiated variables in the definition formula. These concept definitions obtained are, then, overly specific.

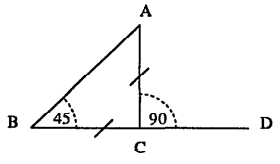

Figure 3: Isosceles Right Triangle Training Example

As an example, consider the following problem from the domain of plane geometry. Given a situation as in Figure 3, we wish to show that if the measure of angle ACD is 90°, then the measure of angle ABC is 45°. This training instance is a particular case of the more general goal concept *Measure(base,val)*, that the measure of the base angle of an isosceles triangle has some value; this goal would arise as a

subgoal to an EBL system that is trying to prove the interesting theorem that any inscribed angle of a circle has half the measure of its intercepted arc.

Suppose our plane geometry domain theory contains, among others, the following facts (where *Minus* (x,y,z) and *Div* (a,b,c) are procedurally defined to be true when $z = x-y$ and $c=\frac{a}{b}$, respectively):

Supp $(ax,ay) \wedge$ *Measure* $(ax,max) \wedge$ *Minus* $(180,max,may)$
$\rightarrow$ *Measure* (ay,may) (1)

Supp $(ax,ay) \wedge$ *Measure* $(ax,90) \rightarrow$ *Measure* $(ay,90)$ (2)

Isos $(tri) \wedge$ *Vertex–Ang* (tri,ang)
$\wedge$ *Measure* $(ang,90) \rightarrow$ *Isos–Right* (tri) (3)

Isos–right $(tri) \wedge$ *Vertex–Ang* (tri,ang)
$\rightarrow$ *Measure* $(ang,90)$ (4)

Isos–Right $(tri) \rightarrow$ *Isos* (tri) (5)

Isos–Right $(tri) \wedge$ *Measure* $(ang,90)$
$\rightarrow$ *Vertex–Ang* (tri,ang) (6)

Isos $(tri) \wedge$ *Vertex–Ang* $(tri,ver) \wedge$ *Base–Ang* (tri,bas)
$\wedge$ *Measure* $(ver,mver) \wedge$ *Minus* $(180,mver,diff)$
$\wedge$ *Div* $(diff,2,mbas) \rightarrow$ *Measure* $(bas,mbas)$ (7)

Isos–Right $(tri) \wedge$ *Base–Ang* $(tri,bas) \rightarrow$ *Measure* $(bas,45)$ (8)

Here, rule 2 is a compiled version of rule 1; in particular, the variable *max* of rule 1 is instantiated with the value 90, evaluation of *Minus* $(180,90,may)$ is performed, and rule 2, stating that the supplement of a 90° angle is itself 90°, is created. In addition, rule 8, stating that the base angle of any isosceles right triangle is 45°, is a compiled version of rule 7 (which applies to all isosceles triangles) with the help of rule 4, rule 5, and the additional knowledge (not listed above) that all isosceles triangles have base and vertex angles. Suppose we are given the following (training instance) information:

Isos (Tri) — *Supp* (ACD,ACB)
Vertex–Ang (Tri,ACB) — *Measure* $(ACD,90)$
Base–Ang (Tri,ABC)

If the procedurally defined predicates are unconditionally operational and we define any concept, or specialization thereof, of the following form to be operational:

Supp $(a_1,a_2) \wedge$ *Measure* $(a_1,ma_1) \wedge$ *Isos* (tr)
$\wedge$ *Vertex–Ang* $(tr,a_2) \wedge$ *Base–Ang* (tr,a_3)

then one possible generalized explanation structure (using compiled rules 2 and 8) for *Measure* $(ABC,45)$ is that in Figure 4. We choose the particular operationality condition above so that we might generate a theorem which calculates the measure of the isosceles triangle's base angle in terms of the angle which is supplementary to its vertex angle.

Unfortunately, the use of compiled knowledge yields an overly specific concept definition. In English, the conjunction of leaf nodes in Figure 4 state the rule that, for any isosceles triangle, if its vertex angle is supplementary to a right angle, then its base angle will be 45°. Thus, the only generalization that took place was from the specific isosceles right triangle of the training instance to all isosceles right triangles.

Even applying the IMEX implication algorithm seems to be of no use initially. The only rule that can be expanded (the only one with all its antecedents below the boundary of operationality) is compiled rule 2; once expanded (essentially

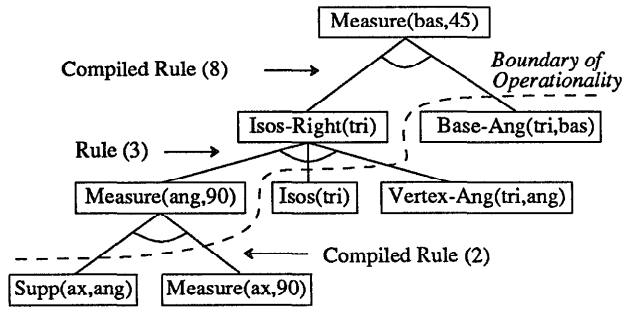

Figure 4: Generalized Explanation Structure of Measure(ABC,45)

replacing rule 2 with rule 1), the concept definition will not get more general because the presence of compiled rule 8 requires the presence of rule 3, which, itself, requires that *Measure* $(ang,90)$ be part of the explanation structure, which, in turn, causes *Measure* $(ax,90)$ to be a leaf of the structure even after rule 2 is expanded. After expanding rule 2, the node *Measure* $(ax,90)$ would be connected to the explanation structure with the antecedent node *Measure* (ax,max) of rule 1. Thus, instead of having *Measure* (ax,max) as a leaf node, *max* would be unified/instantiated with the value 90: the concept definition is overly specific because of an unnecessary variable instantiation.

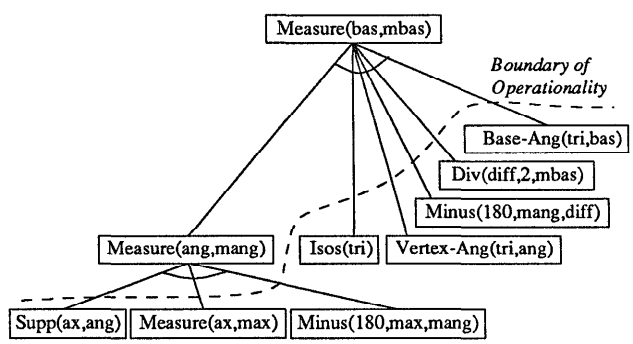

Figure 5: Generalization Possible After Expanding Compiled Rule 8

However if, as in Figure 5, we expand compiled rule 8 (replacing it with rule 7) along with compiled rule 2, then we may eliminate rule 3 from the explanation structure and get the general, operational, desirable rule:

Supp $(ax,ang) \wedge$ *Measure* $(ax,max) \wedge$ *Minus* $(180,max,mang)$
$\wedge$ *Isos* $(tri) \wedge$ *Vertex–Ang* (tri,ang)
$\wedge$ *Base–Ang* $(tri,bas) \wedge$ *Minus* $(180,mang,diff)$
$\wedge$ *Div* $(diff,2,mbas) \rightarrow$ *Measure* $(bas,mbas)$

In English, this states that for all isosceles triangles, the measure of the base angle is half the measure of the angle which is supplementary to the triangle's vertex angle. The key problem is to keep from having to expand all of the compiled rules which appear above the boundary of operationality when trying to generalize explanation structures like the above.

3.2. The IMEX Instantiation Algorithm

Due to space constraints and the complexity of the method, we will only briefly sketch how to handle overly instantiated concept definitions. After compiled knowledge has been used to generate an initial explanation structure, the IMEX Implication Algorithm should be run on the resultant structure to generalize it as much as possible. Next, for each leaf node of the resulting explanation structure, if the leaf node is more specific (in terms of more variable instantiations) than the corresponding uninstantiated antecedent node of the rule that links the leaf to explanation structure, then do the following: Trace up the explanation structure from the leaf until the antecedent of a compiled rule is found. Expand this rule, retracting the unification constraints resulting from the connections between its old specific antecedents and adding the new constraints from its new more general antecedents. Check to see if the propagation of these constraints generalizes the leaf node sufficiently. If so, then we are done with that leaf node. Otherwise continue tracing up the proof structure to find more compiled rules to expand.

4. Discussion and Future Work

Both IMEX algorithms rely on being able to expand compiled knowledge. This expansion process can be made more efficient if the justifications for the knowledge are recorded when the knowledge is compiled. Otherwise, these justifications must be redetermined for each expansion step. If the original domain theory contains recursive domain rules, then it is possible for recursive pieces of compiled knowledge to be generated. Thus, any implementation of the IMEX algorithms must include some type of goal stack checking to avoid getting into infinite loops while expanding rules.

In addition, IMEX must be capable of computing the boundary of operationality in order to direct its search through the space of possible explanation structures. Braverman and Russell [1988] give methods for finding the boundary and describe a number of properties of operationality theories that affect the ease with which the boundary may be found. If the operationality theory satisfies a property termed *locality*, then the new boundary in step (3) of the implication algorithm may be obtained by only modifying the old boundary in the region of the newly spliced-in rule expansion. With other types of operationality theories, especially those which allow predicates to be conditionally operational, finding the boundary can be more complex; in fact, more than one boundary may exist, leading to concept definitions that are mutually incomparable along the generality/specificity dimension. Choosing between these different boundaries is a matter for future research.

IMEX only attempts to maximize the generality of the concept definition based on an initial explanation structure. In a sufficiently complex domain there may be several significantly distinct explanation structures that explain the training instance (such as proving, if possible, *Safe-To-Stack(x,y)* in terms of the *Not(Fragile(y))* rule as opposed to the *Lighter(x,y)* rule). In the future, we would like to investigate methods of finding the most general concept definition achievable considering as many of those structures as is feasible.

Currently, we are in the process of implementing a system which applies the IMEX method in the domain of route planning. The creation and use of compiled knowledge effectively allows for reasoning by levels of abstraction. We believe that IMEX, in conjunction with other processes for removing and reordering rules, will be able to efficiently approximate the kind of optimal levels of abstraction proposed by Korf [1987]. Our goal is to create a system whose global performance converges to approximate optimality via local improvements in the domain theory.

References

[Bennett, 1987] Scott W. Bennett. Approximation in mathematical domains. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (pp. 239-241). Milan, ITALY: Morgan Kaufmann, August 1987.

[Braverman and Russell, 1988] Michael S. Braverman and Stuart J. Russell. Boundaries of Operationality. In *Proceedings of the Fifth International Conference on Machine Learning*. Ann Arbor, MI: Morgan Kaufmann, June 1988.

[Cullingford, 1978] Richard E. Cullingford. Script application: Computer understanding of newspaper stories: Yale University Computer Science Research Report #116, 1978.

[Dejong and Mooney, 1986] Gerald F. Dejong and Ray Mooney. Explanation-based learning: An alternative view. *Machine Learning, 1(2)*, 145-176, 1986.

[Ellman, 1988] Tom Ellman. Approximate theory formation: An Explanation-based approach. In *Proceedings of the Seventh National Conference on Artificial Intelligence*. St. Paul, Minnesota: Morgan Kaufmann, August 1988.

[Fikes, Hart, & Nilsson, 1972] Richard E. Fikes, Peter E. Hart, and Nils J. Nilsson. Learning and executing generalized robot plans. *Artificial Intelligence, 3(4)*, 251-288, 1972.

[Hirsh, 1987] Haym Hirsh. Explanation-based generalization in a logic-programming environment. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (pp. 221-227). Milan, ITALY: Morgan Kaufmann, August 1987.

[Hirsh, 1988] Haym Hirsh. Reasoning about operationality for explanation-based learning. In *Proceedings of the Fifth International Conference on Machine Learning*. Ann Arbor, MI: Morgan Kaufmann, June 1988.

[Keller, 1987] Richard M. Keller. Defining operationality for explanation-based learning. In *Proceedings of the Sixth National Conference on Artificial Intelligence* (pp. 482-487). Seattle, WA: Morgan Kaufmann, July 1987.

[Korf, 1987] Richard E. Korf. Planning as search: A quantitative approach. *Artificial Intelligence, 33*, 65-88, 1987.

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

[Mitchell *et al.*, 1986] Tom M. Mitchell, Richard M. Keller, & Smadar T. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning, 1(1)*, 47-80, 1986.

[Mostow, 1987] Jack Mostow. Searching for operational concept descriptions in BAR, MetaLEX, and EBG. In *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 376-389). Irvine, CA: Morgan Kaufmann, June 1987.