Vivid Knowledge and Tractable Reasoning: Preliminary Report

David W. Etherington
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974-2070

Alex Borgida
Department of Computer Science
Rutgers University
New Brunswick, NJ

Ronald J. Brachman
Henry Kautz
AT&T Bell Laboratories
Murray Hill, NJ 07974-2070

Abstract

Mundane, everyday, reasoning is *fast*. Given the inherent complexity of sound and complete reasoning with representations expressive enough to capture what people seem to know, commonsense reasoning must require shortcuts and assumptions. Some means of simplifying the retrieval of the inferential consequences of a set of facts is obviously required. Instead of looking, as others have, at limited inference or syntactic restrictions on the representation, we explore the use of "vivid" forms for knowledge, in which determining the truth of a sentence is on the order of a database retrieval.

In order to base a reasoning system on vivid knowledge, we consider ways to construct a vivid KB—a complete database of ground, atomic facts—given facts that may be presented in a more expressive language that allows incompleteness (e.g., first-order logic). Besides offering an architecture for examining these problems, our results show that some forms of incomplete knowledge can still be handled efficiently if we extend a vivid KB in a natural way. Most interesting is the way that this approach trades accuracy for speed.

1 Introduction

People perform quickly and competently in most everyday situations—despite an overwhelming barrage of information that nonetheless does not unambiguously characterize the state of the world. In contrast, computer problem-solvers—especially those with clear, formal foundations—are extremely slow in most circumstances, even when presented with little information.

Consider a problem-solver that relies on a knowledge representation (KR) system to answer queries about what follows from a knowledge base. Although there are many factors that contribute to the overall performance of the problem-solver, clearly the efficiency of the KR system is important. Recent attempts to deal with the intractability of such systems have generally fallen under two headings: limited languages (e.g., [Patel-Schneider 1984, Borgida *et al* 1989]), and limited inference (e.g., [Frisch 1988, Patel-Schneider 1989]). In the former, what can be expressed in the knowledge base (KB) is restricted (sometimes severely) to guarantee that queries can be answered in more or less reasonable time. In the latter, restrictions like avoiding chaining or four-valued interpretations yield limited conclusions, albeit from relatively expressive KB's.

We conjecture that a key to efficient problem-solving lies in a notion of *commonsense reasoning*—the kind of reasoning that people engage in all the time without recourse to "paper and pencil", reasoning by cases, backtracking, or particularly deep thought.[1] Commonsense reasoning is fast: if it were a problem-solver's normal mode of reasoning, then the problem-solver would be fast. Paradoxically, studies of commonsense reasoning in AI (e.g., nonmonotonic logics) have frequently led to mechanisms that are even less tractable than logical deduction.

This paper describes an attempt to bridge the gulf between principled theories of inference and practical inference systems. We discuss some components that might combine to support fast reasoning, and a uniform architecture that incorporates them. Obviously, commonsense reasoning is inherently approximate and fallible. Our architecture lets us move towards commonsense performance, and yet still say something substantive about the system's relationship to "ideal competence".

2 Vivid Reasoning

What would be a good basis on which to build a fast reasoner? (Given the massive amounts of information agents are faced with, we cannot even interpret "fast" as "polynomial-time"—we really need performance sublinear in the total size of the KB for simple queries.) The natural candidate from Computer Science is something like a relational database, where query-

[1] We call reasoning that does not fit this description *puzzle-mode* reasoning, after logic puzzles of the form "The man who owns the camel lives next to the orange-juice drinker...".

answering/reasoning is merely look-up for the kinds of simple questions that we expect to be frequently asked of the KB.

Analyses such as Levesque's [1986] and Reiter's [1984] suggest that a crucial factor in the efficiency of databases is the assumption that the database has a complete and accurate view of the world. Generalizing from this, we conjecture that the proper basis for commonsense reasoning is some "vivid" representation of knowledge—one that bears a strong and direct resemblance to the world it represents. A vivid representation has symbols that stand in a one-to-one correspondence to objects of interest in the world, with connections between those symbols corresponding to relationships of concern. For example [Levesque 1986], a KB containing the sentences "Dan drank 7 ounces of gin" and "Jack drank 6 ounces of gin" would be vivid, with respect to the amount Jack and Dan drank individually, while one containing "Jack and Dan together polished off 13 ounces of gin" and "Dan had one more 1-ounce drink than Jack" would not, despite the fact that the same information follows from both KB's.

The notion of vivid representations is appealing for reasons beyond supporting reasoning as database-style lookup: it corresponds well to the kind of information expressed in pictures; thus, it is reasonable to think that much of the information we gain (i.e., perceptually) occurs naturally in vivid form. Also, various psychologically-oriented explanations of cognition suggest that people often seem to reason directly from "mental models" [Johnson-Laird 1983], rather than by syntactic manipulation of sentential constructs.

Of course, not all information we obtain about the world is in vivid form: linguistic communication, for example, may yield disjunctive or otherwise incomplete or general input (e.g., "Joe doesn't have his Ph.D. yet." or "Everyone in the department has an advanced degree."). Fortunately, much of this information can be coerced into a vivid form in a principled way.

3 System Architecture

What is needed is an appropriate architecture that would allow an AI system to fall back on more general reasoning (e.g., first-order logic) when necessary, but would depend primarily on efficient, vivid reasoning. The approach of standard "hybrid" reasoners, which delegate questions to submodules that can handle them efficiently will not suffice. We need a much more active approach, in which incoming information is processed to augment and maintain a vivid view of the world. We have been investigating an architecture that exemplifies this view (see Fig. 1): first-order facts are "vivified" into a knowledge base of a special form (the VKB).[2] This vivification may lose information, since the VKB cannot *explicitly* represent disjunction, negation, or any form of incompleteness. This makes it important to determine the relationship between the answers that a complete theorem-prover would return when queried, given the KB, and the answers that would be retrieved from the vivid knowledge in the VKB—i.e., between α and α' in the figure. This relationship can be thought of as the degree of soundness and completeness of the VKB. Vivid reasoning will not be very useful if α' is too small a subset of, or bears no understandable relationship to, α.

Figure 1: Simple view of a vivid knowledge base.

Because not all reasoning fits our commonsense reasoning paradigm, we propose a hybrid system that retains the original information to supplement, as necessary, the vivid form. We attempt to answer queries by simple retrieval directly from the vivid KB. If that provides inadequate answers, general or special-purpose reasoning with the original KB may be tried, perhaps depending on the importance of the query. Ultimately, one measure of success will be the proportion of reasoning that can be delegated to the VKB.

Figure 2: A more general architecture.

A generalization of the architecture of Fig. 1, and a more realistic view, is illustrated in Fig. 2. Notice first

[2] We distinguish below between the KB—the knowledge given to the system—and the VKB—the system's vivid representation of that knowledge.

the influence of a variety of components on the vivification of the original facts. Universal rules affect vivification simply and directly (see below). However, where the available knowledge is incomplete, we can often do better than simply leaving the information in non-vivid form. It may be possible, for example, to eliminate the ambiguity of the given disjunction in favour of definite facts—facts not strictly equivalent, but sufficient for the purposes of the system. For example, defaults or preferences[3] can be used to capture the contribution of previous experience, "Gricean" communication conventions [Grice 1975],[4] and linguistic context effects in forming mental models. In other cases, abstraction provides a powerful tool. In some circumstances, it may even suffice to make arbitrary choices, as suggested in Levesque's *Computers and Thought* lecture. Thus, the information in the VKB may be the consensus of multiple knowledge sources, as suggested by Fig. 2.

It also seems useful to separate out parts of the original KB that are essentially taxonomic. As we show below, taxonomies provide another form of disjunctive information that can be used efficiently in vivification and retrieval.[5]

4 Constructing a Vivid Knowledge Base

A vivification process for the simplest case—that of non-disjunctive, positive (possibly universally quantified) sentences—is easy to imagine. All that is necessary is to take the set of instances of the universally-quantified formulae over the set of known individuals and store the result as a collection of positive, ground, atomic predicates (e.g., a relational database). However, we also intend to take information that would appear suitable only for the KB, and use it in vivification and/or in conjunction with the VKB in query-answering.

The architecture described above trades effort and space as knowledge is added to the system in favour of rapid query-answering. Although there are fall-back positions that make vivification less demanding, some of which are discussed below and in [Borgida & Etherington 1989], it is useful to ignore the cost of vivification at first, to make some of the underlying theoretical issues more apparent. Notice, however, that vivification is not the same as computing all consequences of the KB: only ground atomic consequences are developed. Furthermore, any ground consequences that can be obtained by database techniques (e.g., membership in defined relations) need not be computed.

Disjunctive and negative information do not fit readily into the database world-view, and are major contributors to the complexity of logical reasoning. We address disjunction and negation piecemeal, distinguishing several different forms and treating each differently. Above all, we strive to avoid reasoning by cases. We hypothesize that commonsense reasoning achieves its efficiency, in part, by not resorting to case analysis, and we treat problems that absolutely require reasoning by cases as puzzle-mode problems.

Perhaps the best way to discuss the various versions of vivification is to consider progressively weaker restrictions on the forms of negative and disjunctive information that can be vivified, and consider how each new class of facts can be converted into vivid form.[6]

4.1 A Simple Case

The simplest extension beyond ground and universally-quantified atoms is to allow disjunctions of the form $\forall x.\ \neg A(x) \vee B(x)$ (equivalently, simple implications like those found in inheritance hierarchies). The vivification algorithm treats these by asserting $B(\alpha)$ whenever $A(\alpha)$ is entered into the VKB. The VKB is then queried as a normal relational database, with negation determined by the *closed-world assumption* (CWA) [Reiter 1978].

For example, vivifying the KB, $\{Man(Socrates)$, $Woman(Ophelia)$, $\forall x.\ Man(x) \supset Mortal(x)\}$, results in the VKB, $\{Man(Socrates)$, $Woman(Ophelia)$, $Mortal(Socrates)\}$. The query $Mortal(Socrates)$ is answered by lookup in the VKB, and returns 'Yes'. The query $Mortal(Ophelia)$ fails in the VKB, so the CWA sanctions the answer 'No'.

The KB's considered so far correspond to *definite databases*, i.e., databases of clauses each containing exactly one positive literal. Reiter [1978] shows that the CWA is always consistent with definite databases. We have proved that the answers returned by closed-world querying of the VKB are identical to those returned by closed-world querying of the original knowledge base under the "domain-closure assumption" (DCA) [Reiter 1978].[7]

Because negative information is not explicitly represented in the vivid KB, it is not necessary to consider the contrapositive forms of the disjunctive rules; any rule not instantiated by the vivification process will be correctly instantiated by the CWA during query-answering. This ensures that the computational complexity of the

[3] At the moment, we assume the defaults are presented to the system in the same declarative way as other facts; eventually, defaults should be created by inspecting the VKB (i.e., from experience).

[4] For example, when someone says, "Some of the chemists are beekeepers," they typically mean to imply that some of them are not [Johnson-Laird 1983].

[5] Interestingly, mathematicians and computer scientists have independently studied "vivid" representations of partial orders, where transitive relationships can be directly "read off" the representation (*viz* [Agrawal *et al* 1989]).

[6] To simplify the rest of our discussion of vivification, we restrict ourselves to monadic predicates. In some cases, this hides only messy details. In others, some details remain to be worked out. Readers are welcome to make whichever assumption their credulity allows.

[7] The DCA, which says that the individuals mentioned by the theory constitute the entire set of individuals, is used in database theory to facilitate handling quantified queries.

vivification process does not get out of hand. In particular, it is not necessary to reason by cases, since the negative case can never be explicitly asserted in the KB. Any technology suitable for reasoning with monotonic semantic networks (e.g., [Thomason *et al* 1987]) can be used to vivify the KB. Of course, the system described so far is not significantly more useful than a monotonic semantic network. In the following sections, we discuss extensions that move in the direction of a useful commonsense reasoning system.

4.2 A Slightly More Complicated Case

The knowledge presented to a system sometimes contains bona fide alternatives and provides no means for deciding amongst them. It is sometimes possible to trade the given ambiguity for vagueness and thereby avoid disjunction. That is, a list of alternatives concerning an individual can sometimes be replaced by a less-fine-grained, but atomic, description that subsumes the alternatives. For example, if we are told only that Joe is 52 or 53, we might represent the fact that Joe is in his early 50's.

To substitute vagueness for ambiguity, we assume that the given KB provides certain subsumption information. This may range from the extreme of a complete upper semilattice containing a subsuming predicate for every subset of the set of predicates (e.g., Fig. 3), through more natural taxonomic hierarchies (e.g., Fig. 4), to the trivial case where everything is subsumed only by $Thing$.

Figure 4: Fragment of a subsumption hierarchy.

In the simplest case amenable to substitution, the given information asserts that a particular individual is a member of one of n classes (i.e., has one of n properties), without specifying which (e.g., $Teacher(Joe) \vee Professor(Joe)$). If the information available in the KB provides a class that subsumes all the mentioned classes, vivification simply asserts membership in the subsuming class, and discards the alternatives, thus obtaining an atomic fact that can be stored in the VKB (e.g., using Fig. 4, $Instructor(Joe)$).

The price of this substitution depends on the density of the available subsumption information. If the subsumption hierarchy is complete, no information is lost: anything deducible from the KB will follow from the VKB. In what we expect to be the more common case of a relatively sparse hierarchy, a certain amount of precision may be lost. Exactly how much will depend on how "natural" the given disjunction is.[8] Disjunctions that are useful for commonsense reasoning will often be subsumed by predicates nearby in the hierarchy. Less natural disjunctions—requiring reasoning closer to puzzle mode—would be subsumed only by much more general concepts—concepts that also subsume many other concepts not represented in the original disjunction.

For example, think again of the hierarchy in Fig. 4. The information that Joe is a professor or a doctor would be vivified by asserting $Professional(Joe)$, allowing the possibility that he is a teacher, a lecturer or a lawyer. Being told that he is a professor or a student would yield $Adult(Joe)$, losing (among other things) the fact that he is not a visitor. Learning that Joe is a lawyer or a shark might give rise only to $Thing(Joe)$.

"Unnatural" disjunctions do not slow vivification down appreciably, but uselessly vague answers can be expected concerning the subjects of these disjunctions. This coincides with our intention that the vivid reasoning component should not be expected to handle puzzle-mode problems well.

We have not said exactly how the VKB should handle negation in this extended representation scheme. In particular, since the KB is no longer definite, it is inappropriate simply to use the CWA, which may introduce inconsistency. For example, if $Teacher(Joe) \vee Professor(Joe)$ is made vivid by representing only $Instructor(Joe)$, then the CWA would justify both $\neg Teacher(Joe)$ and $\neg Professor(Joe)$, since neither follows from $Instructor(Joe)$.

One solution is not to make the CWA at all; then failure to find a fact in the VKB would simply mean that the VKB didn't know the fact to hold. This solution seems a bit radical, however. While avoiding overcommitment when given vague knowledge, it prevents making the CWA even for things not even represented in the KB. This would make the VKB much less vivid. Fortunately, there is a less drastic solution.

[8] We realize, of course, that "natural" is not a well-defined term. In this context, however, we can define a disjunction as *natural* if its elements are subsumed by a predicate nearby in the abstraction hierarchy. We can justify this name by begging the question: we assume that those responsible for building the KB will include nodes for natural disjunctions of concepts, and not for unnatural ones!

Figure 3: Complete structure for {*Teacher, Lecturer, Professor*}.

For KBs of the form we are considering, the appropriate form of the CWA is the Generalized CWA (GCWA) [Minker 1982], which is much like the CWA, except that it avoids asserting the negation of terms involved in irreducible disjunctions. It turns out to be straightforward to augment the representation mechanism used in the VKB to allow it to distinguish "unknown by virtue of no information" from "unknown by virtue of vagueness". The CWA can then be applied in the extended representation to infer the negations of terms for which no information is available. We have shown [Borgida & Etherington 1989] that this approach yields the same results as the GCWA applied to the original KB, assuming a complete subsumption hierarchy. A sparse hierarchy, of course, may result in weaker statements, due to the loss of precision in the construction of the VKB.

The representation and algorithms we have developed are particularly attractive because they have the property that their accuracy degrades gracefully as their efficiency improves, and does not degrade for unambiguous information. Thus the retrieval algorithms are sound and complete in cases where the hierarchy is complete or where the given knowledge either is atomic or corresponds to concepts directly representable in the hierarchy. In exchange for the loss of representational fidelity in other cases, we achieve significant performance improvements: assuming the hierarchy has $O(p)$ predicates, where p is the number of primitive predicates, query-answering is *sublinear* in the number of facts told to the KB, and linear in the size of the query. Complete query-answering, on the other hand, is at least $O(n \log n)$ in the size of the query, and linear in the size of the KB. We can also achieve significant improvements in the complexity of telling the KB facts: the NP-complete problem of converting inputs to conjunctive normal form suitable for vivification can be approximated, without additional loss of information, in polynomial time.

4.3 A Still More Complicated Case

Another natural-seeming form of disjunction involves alternation of the same predicate over more than one individual, e.g., $Teacher(Joe) \vee Teacher(Bill)$. We treat these using a technique similar to that discussed in the previous section, abstracting a set of individuals to a type containing them. In this case, a disjunction is vivified by introducing a Skolem constant (a null value, in database terminology) to represent whichever individual satisfies the predication. We assert that the predicate holds of the null, and that the null is a member of the appropriate type.[9] As in the predicate case, the information lost by vivifying this way is proportional to the density of the type hierarchy. The empirical question of whether there will generally be types available that cover enough of the disjunctions over small sets of individuals (especially sets of two) that occur in commonsense reasoning remains open. We can construct intuitively plausible arguments that there will be no problem, but we have not yet compiled data.

It *is* possible that the hierarchy will be too sparse, in which case some combination of the above technique with another we are studying may be necessary. This second approach maintains a list of constraints on which individuals may replace the null. These constraints can be used in a variety of ways to provide answers of varying detail. In the limit case, a general-purpose equality reasoner can be applied to recover all the information inherent in the original problem statement. At the other extreme, it is possible to manipulate a unification algorithm in such a way that it provides maximally optimistic or pessimistic views of the effects of the constraints on answers to queries, while performing only a fraction of the work necessary to determine the "correct" (maximally informative) answer. This allows the system to quickly provide upper and lower bounds [Lipski 1979]

[9] The same techniques can be applied directly to range-limited existentially-quantified statements, such as $\exists x/Teacher.\ Salary(x) > \$40,000$.

on the answers to queries while avoiding the effort necessary to determine the maximally-informative answer.

Again, the approximations (which are motivated by the structure imposed on the world by the hierarchy) made in vivifying the KB result in gratifying performance gains like those discussed in the previous section, while retaining soundness and completeness where possible. In this case, the gains are even more significant, since the general query problem for the class of formulae treated here is NP-complete in the size of the query [Imielinski 1988]. Our algorithms can approximate answers in polynomial time.

4.4 Negation

The major deficiency in the system as it stands is that it provides no mechanism for explicitly telling the VKB negative information. For many applications, however, this is not a particular problem, since the system has the GCWA to provide implicit negation. In fact, this is no less than is provided by many AI knowledge-representation systems (e.g., PROLOG). Occasionally, however, it would be useful to have explicit negation. While we are still working out the details, it appears to be simple to add capabilities for representing simple ground atomic negative facts, and perhaps uniformly negative ground disjunctive facts like the uniformly positive disjunctions discussed above (e.g., $\neg Teacher(Joe) \vee \neg Professor(Joe)$). Such an extension would allow the system to distinguish between "definitely false", "false by the CWA", and "unknown", in some cases.

4.5 Life in the Space/Time Continuum

In describing our vivification algorithm, we have been profligate with the space and, to a lesser extent, time required to represent knowledge, in an effort to achieve optimal performance for query-answering. Depending on the availability of storage and the relative frequencies of update and query, it may be desirable to retreat from the extreme position of a totally vivid representation. Since the architecture of our hybrid system assumes that the original KB is available for the use of the problem solver, it is not particularly difficult to beat this retreat.

The hierarchies that we use are exception-free, and support efficient computation. Provided certain conventions are followed during vivification (and update), it is easy to eliminate some of the worst space consumption. In particular, the transitive closure of the inheritance hierarchy need not be computed in advance, and the distinction between "unknown" and "assumed false" can be determined as required to answer queries, rather than explicitly stored.

5 Knowing, More or Less?

The techniques we have described for vivifying a KB result in the system knowing less (at least no more) than it was told. Reasoning by cases is avoided by only representing versions of the input that can be made unambiguous. Levesque [1986] suggests that people sometimes avoid the expense inherent in disjunctive information by simply picking one disjunct. He argues that a tremendous amount of vague and ambiguous information is presented to an agent all the time and it is often sufficient (or even necessary) to disambiguate it either arbitrarily or according to some default principles, in effect coming to "know" *more* than was told.[10]

Research on default reasoning has concentrated on developing default theories that are epistemically adequate, but has ignored computational complexity. Conversely, we are interested in using defaults to support fast vivid problem solving—including problems that could be solved more slowly without defaults—and are centrally concerned with processing the default rules quickly.

To see how defaults might be used in vivification, suppose a partial description of a room is input to an agent's KB. Some items are precisely specified (e.g., the coordinates of the door), but others are missing (e.g., the width of the fireplace opening). Geometric-level vivification might fill in the missing information with typical or random values. For instance, fireplaces are typically 30″ wide, and the windows could be assigned random positions along the exterior walls. Finally, suppose an agent is told that there is a stack of 24″-long firewood outside, and is instructed to build a fire in the fireplace. She can create a simple plan to carry some firewood indoors and place it in the fireplace because she has made the default assumption that the fireplace opening is larger than the wood. Lacking this belief, she would need to generate a conditional plan: measure the fireplace; if it is less than 24″, then cut the firewood into suitably-sized pieces.

The utility of defaults does not depend on a coordinate-level view of the world. For example, suppose an agent is driving a car when a tire goes flat. Her vivid model of the car has a good inflated tire and a jack in the trunk. This belief, together with the goal of replacing the flat tire, generate the obvious plan: open the trunk, remove the spare tire and jack, raise the car, and replace the flat tire with the spare. She does not create a plan that considers the possibility that there is no spare in the trunk and conditionally sends her off in search of one.

Vivid reasoners never need to generate conditional plans, or reason about the possible ways the world could be. The vivid model forms the basis for a direct solution to the problem at hand. If the solution fails because the default assumptions are incorrect, then the vivid model is revised: the blatantly erroneous assumptions are replaced by new observations, the KB is re-vivified, and problem-solving is repeated.[11]

[10] See [Etherington 1988] for a discussion of the pervasiveness of default reasoning in intelligent behaviour.

[11] Obviously, there are situations in which the problem-solver might choose a more conservative (and expensive)

This use of defaults for vivification is closely related to the "qualification problem" in planning [McCarthy 1977]. In many domains the list of circumstances that would require different solutions to a planning problem is infinite, so that it is impossible even in principle to solve the problem by reasoning by cases.

We have begun the task of integrating defaults into a vivid reasoner by analyzing the complexity of simple default systems. Selman and Kautz [1988] report our analysis of "model preference default rules", which enforce a simple preference relation over the space of models of a theory (which corresponds to the space of vivid models). While finding a most-preferred model is, in general, NP-hard, one can be found in polynomial time if the preference rules are acyclic. Kautz and Selman [1989] extend this analysis to restricted versions of Reiter's default logic. It is almost always difficult to determine whether a fact holds in any or all "extensions" (roughly, the vivid models) of a theory. Fortunately, however, there is a broad class of theories for which one can find *some* extension in polynomial time. These results show that while default reasoning can be surprisingly complex, there is strong hope for finding tractable default vivification algorithms for limited cases.

6 Directions for Future Work

There are many open problems that we intend to explore. Among these are questions about the effects of closed-world reasoning concerning the hierarchies. In particular, we are interested in providing mechanisms for indicating mutual exclusion and exhaustive partitions of classes.

We are also considering the effects of different assumptions when vivifying existentially-quantified formulae. Alternatives include assuming the existentially-specified individual is none of those known to satisfy the specified properties (this corresponds to the Gricean assumption that, since there is no point in telling someone something they already know, seemingly redundant inputs should be assumed to contain new information), completely ignoring existential formulae entailed by what is known, and making domain-closure assumptions vis-à-vis existential quantifiers (assuming their referents are among the known individuals). We suspect that it may be necessary to allow knowledge sources to control such aspects of vivification explicitly, presumably augmented by suitable default choices, much as relational database technology provides for a variety of kinds of null-values to express fine shades of interpretation [Codd 1979].

7 Conclusions

We have outlined an architecture for a KR system that supports efficient treatment of commonsense reasoning problems. The essential idea is to use an array of techniques to transform information about the world, which may be incomplete, into a vivid representation in which inference approaches simple inspection of the representation. By trading representational fidelity for speed, we are able to achieve attractive performance in certain situations. In any case, the loss of accuracy can be motivated, predicted, and controlled by decisions made as knowledge is presented to the system.

approach.

Acknowledgements

We are grateful to Hector Levesque and Bart Selman for their contributions to the work described here.

References

Agrawal, R., Borgida, A., and Jagadish, H. V. [1989], "Efficient Management of Transitive Relationships in Large Data and Knowledge Bases", *Proc. ACM-SIGMOD–89*, Portland, OR, May 1989.

Borgida, A., and Etherington, D. W. [1989], "Hierarchical Knowledge Bases and Tractable Disjunction", *Proc. First International Conference on Knowledge Representation and Reasoning*, Toronto, Canada, May, 1989, 33-43.

Borgida, A., Brachman, R. J., McGuinness, D. L., and Resnick, L. A., "CLASSIC: A Structural Data Model for Objects", *Proc. ACM-SIGMOD–89*, Portland, OR, May, 1989.

Codd, E. F. [1979], "Extending the Database Relational Model to Capture More Meaning", *ACM Trans. on Database Systems* 4(4), December, 1979, 397-434.

Etherington, D. W. [1988], *Reasoning with Incomplete Information*, Pitman Research Notes in Artificial Intelligence, Pitman (London)/Morgan Kaufmann (Los Altos), 1988.

Frisch, A. M. [1988], *Knowledge Retrieval as Specialized Inference*, Technical Report UIUCDCS-R-88-1404, Department of Computer Science, University of Illinois at Urbana-Champaign, February, 1988.

Grice, H. P. [1975], "Logic and Conversation", in *Syntax and Semantics, Vol 3: Speech Acts*, Cole, P. and Morgan, J. L. (eds.), Academic Press, 1975.

Imielinski, T. [1988], "Incomplete Deductive Databases", submitted for publication, 1988.

Johnson-Laird, P. N. [1983], *Mental Models*, Harvard University Press, Cambridge, MA, 1983.

Kautz, H., and Selman, B. [1989], "Hard Problems for Simple Default Logics", *Proc. First International Conference on Knowledge Representation and Reasoning*, Toronto, Canada, May, 1989, 189-197.

Levesque, H. J. [1986], "Making Believers out of Computers", *Artificial Intelligence* 30, October 1986, 81-108. (Originally given as the "Computers and Thought" lecture at IJCAI-85.)

Lipski, W. Jr. [1979], "On Semantic Issues Connected with Incomplete Information Databases", *ACM Trans. on Database Systems* 4, 1979, 262-296.

McCarthy, J. [1977], "Epistemological Problems of Artificial Intelligence", *Proc. Fifth International Joint Conf. on Artificial Intelligence*, Cambridge, MA, 1977, 1038-1044.

Minker, J. [1982], "On Indefinite Databases and the Closed-World Assumption", *Proc. Sixth Conf. on Automated Deduction*, New York, June, 1982, Springer-Verlag, NY.

Patel-Schneider, P. F. [1984], "Small can be Beautiful in Knowledge Representation", *Proc. IEEE Workshop on Principles of Knowledge-Based Systems*, Denver, December, 1984, 11-16.

Patel-Schneider, P. F. [1989], "A Four-Valued Semantics for Terminological Logics", *Artificial Intelligence* 38, 1989, 319-351.

Reiter, R. [1978], "On Closed-World Data Bases", in Gallaire, H. and Minker, J. (eds), *Logic and Data Bases*, Plenum Press, 1978, 55-76.

Reiter, R. [1984], "A Logical Reconstruction of Relational Database Theory", in Brodie, M. L., Mylopoulos, J., and Schmidt, J. W. (eds), *On Conceptual Modelling*, Springer-Verlag, 1984, 191-233.

Selman, B. and Kautz, H. [1988], "The Complexity of Model-Preference Default Theories", *Proc. Canadian Soc. for Computational Studies of Intelligence-88*, Edmonton, Alberta, June, 1988, 102-109.

Thomason, R., Horty, J., and Touretzky, D. [1987], "A Calculus for Inheritance in Monotonic Semantic Nets", in *Methodologies for Intelligent Systems*, Ras, Z. and Zemankova, M., eds., North-Holland, Amsterdam, 1987, 280-287.