

Toward Efficient Default Reasoning

David W. Etherington and James M. Crawford

Computational Intelligence Research Laboratory

1269 University of Oregon

Eugene, OR 97403-1269

{ether, jc}@cir1.uoregon.edu

Abstract

Early work on default reasoning aimed to formalize the notion of quickly “jumping to conclusions”. Unfortunately, the resulting formalisms have proven more computationally complex than classical logics. This has dramatically limited the applicability of formal methods to real problems involving defaults. The complexity of consistency checking is one of the two problems that must be addressed to reduce the complexity of default reasoning. We propose to approximate consistency checking using a novel synthesis of limited contexts and fast incomplete checks, and argue that this combination overcomes the limitations of its component parts. Our approach trades correctness for speed, but we argue that the nature of default reasoning makes this trade relatively inexpensive and intuitively plausible. We present a prototype implementation of a default reasoner based on these ideas, and a preliminary empirical evaluation.

Computation and Nonmonotonicity

Early work on nonmonotonic reasoning (NMR) was often motivated by the idea that defaults should make reasoning easier. For example, Reiter (1978) says “[closed-world reasoning] leads to a significant reduction in the complexity of both the representation and processing of knowledge”. Winograd (1980) observes that agents must make assumptions to act in real time: “A robot with common sense would [go] to the place where it expects the car to be, rather than . . . thinking about the infinite variety of ways in which circumstances may have conspired for it not to be there.”

Paradoxically, formal theories of NMR have been consistently characterized by their intractability. For example, first-order default logic (Reiter 1980) is not semi-decidable and its inference rules are not effective. In the propositional case, most NMR problems are Σ_2^P or Π_2^P -complete (Gottlob 1992; Stillman 1992).¹ Even very restricted sublanguages based on propositional languages with linear decision procedures remain

NP-complete (Kautz & Selman 1989). Convincing examples of broadly useful theories within demonstrably tractable languages for NMR have yet to appear.

A nonmonotonic formalism sanctions a default conclusion only if certain facts can be shown to be consistent with the rest of the system’s beliefs—i.e., only if it can be shown that the default is not a known exceptional case. Unfortunately, consistency is generally even harder to determine than logical consequence. The need to prove consistency before drawing default conclusions is the first source of the intractability of nonmonotonic formalisms.

The second source of intractability is that the order in which default rules are applied can effect the extension generated. It is these two sources of intractability together that produce the Σ_2^P (or Π_2^P) time complexity of most problems in default reasoning. However, given an oracle for consistency checking, some interesting problems, such as finding an extension for a normal default theory, could be solved tractably. Conversely, an oracle for default ordering would produce tractability only for languages with very limited expressive power. Furthermore, the ability to check consistency quickly is interesting in its own right for many propositional reasoning tasks. Therefore, we believe that a first step toward developing practicable nonmonotonic reasoners is to reduce their dependency on intractable consistency checking.

Our approach to approximate consistency checking is ultimately based on limiting the search for exceptions. This approach has the intuitive appeal that a default can be applied without first having to discount every possible reason this case might be exceptional. We hope to recapture the intuition that a default should be applied unless its inapplicability is readily apparent (i.e., “at the top of your mind”). Our approach trades accuracy for speed: “inappropriate” conclusions may be reached that must be retracted solely due to additional thought, but this tradeoff accords with the original arguments for default reasoning. More importantly, we argue, defaults generally seem to be used in ways that minimize the cost of this tradeoff.

We limit our discussion to default logic (Reiter 1980), but it is important to note that our ideas apply directly to other nonmonotonic formalisms. A default

¹Arguably, this complexity is the price of increased expressivity, allowing NMR formalisms to represent knowledge that can’t be concisely expressed in monotonic logics (Cadoli, Donini, & Schaerf 1994; Gogic *et al.* 1995) but this observation is little help in building practical systems.

has the form $\frac{P(\bar{x}):J(\bar{x})}{C(\bar{x})}$, where P , J , and C are formulae whose free variables are among $\bar{x} = x_1, \dots, x_n$; they are called the *prerequisite*, *justification*, and *consequent* of the default, respectively. The default can be read as saying that things satisfying P typically satisfy C unless known not to satisfy J .

Sufficient Tests for Consistency

Consider testing whether β is consistent with a KB. The good news is that there are fast *sufficient* tests for consistency. For example, provided the theory and β are each self-consistent, it suffices (but is not necessary) that no literal in $\neg\beta$ occurs in the clausal representation of the theory. This can be tested in at worst linear time even for non-clausal theories. Similarly, if $\neg\beta$ occurs only in clauses with pure literals, β is consistent with the KB. More complicated tests derive from techniques in (Borgida & Etherington 1989), knowledge compilation (Selman & Kautz 1991) and multi-valued entailment (Cadoli & Schaerf 1992).

Unfortunately, there are two serious obstacles to using such fast tests. Those fast enough to check the whole KB in real time can be expected to fail in realistic applications. It would be a peculiar KB that, for example, had the default “Birds usually fly” with *no* information about non-flying birds! Representing a rule as a default seems to presume knowledge (or at least the strong expectation) of exceptions. This will cause the fast tests described above to fail, giving no useful information. The more complicated tests, such as knowledge compilation, are too expensive to do on the whole KB before each default is applied. Moreover, since applying defaults expressly changes what is believed, compilation cannot be done once in advance.

Context-Limited Consistency Checking

If computational resources are limited, it makes sense to focus our search for inconsistency on the relevant parts of the KB. For example, the default that you can get a babysitter might fail for prom night, but is unlikely to be affected by the stock market; a limited reasoner that devotes much effort to seeing if market fluctuations prevent hiring a sitter seems doomed.

Focusing on limited contexts provides two benefits. First, in the propositional case, consistency checking can be exponential in the size of the theory (if $P \neq NP$). Clearly, if we need only check a small subset, efficiency will improve significantly. Second, one can use fast consistency checks and limited contexts together to help gain efficiency even in first-order logic, where full consistency checking is undecidable.

Ideally, the context should contain exactly the formulae relevant to determining consistency. Then all necessary information is available, and irrelevant search is curtailed: consistency checking is no harder than it must be for correctness. Of course, this ideal solves the problem by reducing it to the arguably

harder problem of determining relevance. Conversely, using a randomly-chosen context for consistency checking could be expected to produce very cheap consistency checks (since the fastest sufficient tests will be likely to succeed), and still have (marginally) better-than-random accuracy (applicable defaults won't be contradicted, and inapplicable default *might* be detected). Naturally, any realistic context-selection mechanism will fall between these extremes. Additional effort spent on context building can reduce the accuracy lost in focusing on the context: like most approximation schemes, practical context selection involves balance.

Just what a context should contain is an open question, but a rudimentary notion suffices to illustrate the idea (c.f. (Elgot-Drapkin, Miller, & Perlis 1987)). Facts come into the context as they are attended to (e.g., from perception or memory), and exit as they become stale. The context should include ground facts known about the objects under discussion (e.g., *Tweety*) as well as rules whose antecedents and consequents are instantiated by either the context or the negation of the justification to be checked (e.g., if *Penguin(Tweety)* is in the context, checking the consistency of *Flies(Tweety)*, should draw in $\forall x. \text{Penguin}(x) \supset \neg \text{Flies}(x)$). Such a context can be built quickly using good indexing techniques.

This simple notion of context can be elaborated in many ways. Limited forms of rule chaining can be provided if chaining can be tightly bounded. For example, if the KB has a terminological component (c.f. (Brachman & Schmolze 1985)), chains through the type hierarchy might be brought in by treating deductions from terminological knowledge as single ‘rule’ applications. Also, “obvious” related items can be retrieved using Crawford’s (1990) notion of the accessible portion of the KB, Levesque’s (1984) notion of limited inference, or other mechanisms that guarantee cheap retrieval.

The significant feature of our approach is the synergy between the two components: context focuses the consistency check on the part of the KB most likely to contain an inconsistency and, often, can be expected to allow fast sufficient checks to succeed where they would fail in the full KB. Such fast tests can allow context-limited consistency testing to be efficient even in large first-order KBs.

A Simple Example: Consider the canonical default reasoning example:

Robin(Tweety), Penguin(Opus), Emu(Edna) ...

$$\begin{array}{ll} \forall x. \text{Canary}(x) \supset \text{Bird}(x) & (1) \\ \forall x. \text{Penguin}(x) \supset \text{Bird}(x) & (2) \\ \forall x. \text{Penguin}(x) \supset \neg \text{Flies}(x) & (3) \\ \forall x. \text{Emu}(x) \supset \text{Bird}(x) & \\ \forall x. \text{Emu}(x) \supset \neg \text{Flies}(x) & \dots \\ \hline \text{Bird}(x) : \text{Flies}(x) & \\ \text{Flies}(x) & \end{array}$$

where the ellipses indicate axioms about many other kinds of birds and many other individual birds. To conjecture that Tweety flies, one must prove $Flies(Tweety)$ is consistent with the above theory—i.e., that $Penguin(Tweety)$, $Emu(Tweety)$, etc. aren't provable. This amounts to explicitly considering all the ways that Tweety might be exceptional, which seems unlike the way people use defaults.

On the other hand, if recent history hasn't brought exceptional types of birds to mind, the context might contain just $Robin(Tweety)$ and (1). A fast test for consistency of $Flies(Tweety)$ would succeed, and so $Flies(Tweety)$ could be assumed. Deciding if Opus can fly, however, brings $Penguin(Opus)$ into the context and hence (2) and (3), so the consistency test fails. Similarly, after a long discussion about various forms of flightless birds, facts about exceptional classes should still be in the context. Fast consistency tests for $Flies(Tweety)$ would thus probably fail, and one would have to explicitly rule out exceptions.

The Mitigating Nature of Defaults

Clearly context-selection is difficult. Fortunately, the nature of defaults makes selection of a useful context less problematic than might be expected. For a default to be reasonable, we contend, (at least) two factors must combine favorably: the likelihood that the consequent holds given that the prerequisite holds and the likelihood that if the prerequisite holds but the justifications are not consistent (so the default is not applicable), the agent will be aware of this fact. If the default is extremely likely to apply, one can tolerate overlooking the odd exception. Similarly, if exceptions are easy to spot, it may be useful to have a default that rarely applies. However, if exceptions are common but difficult to detect, one is ill-advised to make assumptions.² Now, if we characterize a “good default” as one for which the probability is low that the prerequisite holds, the justification is inconsistent, and the inconsistency will not be noticed, we are guaranteed that a context-based system will produce accuracy as good as its defaults.³

Experimental Results

We now turn to the results of preliminary experiments beginning the validation of our approach. A completely convincing test would involve extensive experiments on large, first-order, real-world nonmonotonic KBs, showing significant computational gains and acceptable error rates. Sadly, the intractability of non-

monotonic formalisms seems to have stifled construction of large KBs with defaults; we hope that this work will be a step toward their construction.

Meanwhile, the goal of these experiments is more modest: preliminary determination of the effect of context limitations on the accuracy and cost of consistency checking for randomly-generated propositional theories. Such theories are generally characterized by two parameters: the number of variables and the number of clauses (the length of all clauses is generally taken to be three). For low clause-to-variable ratios, almost all problems are satisfiable, and most problems are computationally easy. At high ratios, almost all problems are unsatisfiable and most problems are easy. In between, in the so-called “transition region”, lies a mixture of satisfiable and unsatisfiable problems, and many quite hard problems (Cheeseman, Kanefsky & Taylor 1991; Mitchell, Selman & Levesque 1992; Crawford & Auton 1993).

Our experiments are primarily in the underconstrained region. In the overconstrained region, almost all theories are inconsistent, so no defaults are applicable. Solving problems in the transition region generally seems to require intricate case-splitting of the kind found more in logic puzzles than in commonsense reasoning. Also, an agent's world knowledge is likely to be fairly underconstrained—we generally know sufficiently little about the world that there are many models that are consistent with what we know.⁴

Working in the underconstrained region, we face a problem: it is likely that a random literal chosen to be our “default” will be consistent with a random theory, and consistency checking in any limited context (even the empty context!) will give the right answer. To solve this problem, we add a randomly generated set of literals to our theories. Intuitively, the clauses correspond to general knowledge about the world and these literals correspond to a set of facts.

The experiments presented below investigate the success of context-limited consistency checking as problem size (# variables, V), degree of constraint (# clauses, C), and number of facts (# literals, L) vary. We find that context limitations are useful in much of the underconstrained region, but their utility drops sharply as we approach the transition region. This is consistent with the generally held belief that, in the transition region, clauses throughout the theory interact with each other in complex ways. Context-limited consistency checking also becomes less useful as L becomes more than about $V/2$; in these cases unit propagation makes full consistency checking so easy that context-limitations become superfluous.

²We ignore the obvious third factor: the cost of errors.

³Gricean principles of cooperative communication seem to enforce the second property above: if the speaker believes the hearer may draw an inappropriate default conclusion from her utterance, she must explicitly block it (Grice 1975), ensuring the appropriate contradiction is in the hearer's context when the default is considered.

⁴Of course, commonsense knowledge no doubt clusters and some of these clusters may be locally quite constraining. Ideally in these cases one would want to choose the context to include the entire cluster, but this goes beyond the scope of the current experiments.

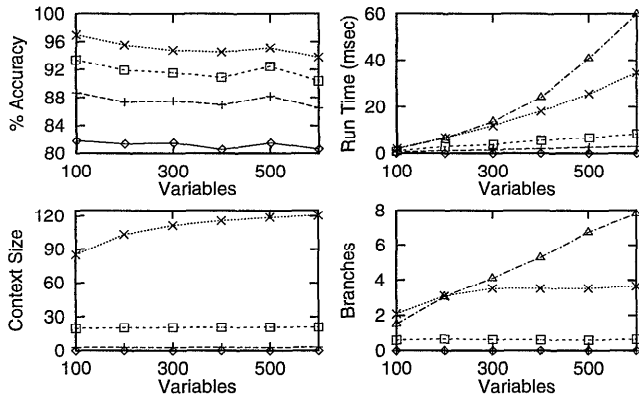


Figure 1: Accuracy, run time, context size, and search tree size, *vs* V , at $L = 0.4V$. $r = 0, 1, 2, 3$, and ∞ are marked \diamond , $+$, \square , \times and \triangle , respectively.

Experimental Setup

We generate random 3-SAT theories using Mitchell *et al*'s (1992) method—each clause is generated by picking three different variables at random and negating each with probability 0.5. There is no check for repeated clauses. Inconsistent theories are discarded. We then randomly select a series of L literals consistent with the theory built so far. Consistency checks are done using TABLEAU (Crawford & Auton 1993).

We select a random literal d to be the “default”, and construct a series of concentric contexts around d . $C_{r,d}$ denotes the context around d with *radius* r . Intuitively, the radius measures how many clauses the context extends out from d . More formally, the context is the subset of the input theory, T , defined as follows: $C_{l,0}$ is $\neg l$ if $\neg l \in T$, and $\{\}$ otherwise. For $r > 0$,

$$C_{r,l} = C_{r-1,l} \cup \bigcup_{x \vee y \vee \neg l \in T} (\{x \vee y \vee \neg l\} \cup C_{\neg x,r-1} \cup C_{\neg y,r-1})$$

(e.g., the $r = 3$ context around l contains the $r = 2$ contexts around l , and around l 's neighbors). TABLEAU is used to test satisfiability. For these tests we modified it to halt the search whenever the current partial assignment satisfies all the clauses in the theory.

Experiment 1: The first experiment tests how the efficacy of context limitation varies with problem size. We varied V from 100 to 600 incrementing by 100. We set C to $2V$ (roughly centered in the underconstrained region), and L to $0.4V$, generated 200 theories, and tested 10 defaults per theory. Each check was done against contexts with radius 0 to 3, and then against the entire theory. The results appear in Figure 1.

The limited growth of the context size is not surprising. A simple probabilistic argument shows that for large problems the expected context size depends on r and C/V (not V). Further, since the number of branches in the search tree depends primarily on the size of the context, it makes sense that the number of branches does not increase appreciably with prob-

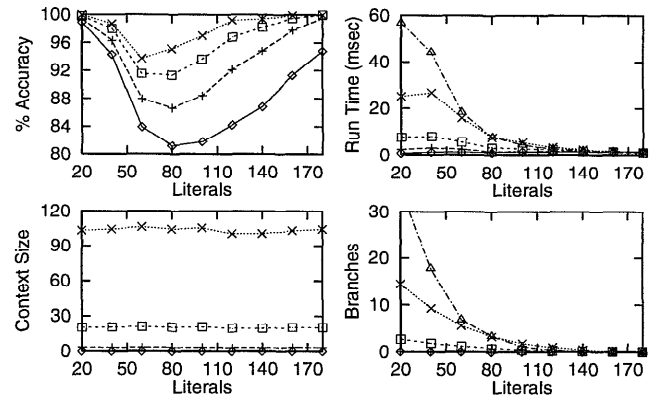


Figure 2: Accuracy, run time, search tree size, and context size, *vs* L , at $V = 200$, $C = 400$. $r = 0, 1, 2, 3$, and ∞ are marked \diamond , $+$, \square , \times and \triangle , respectively.

lem size. Run time depends on the number of branches and on the time spent at each node. However, the time TABLEAU spends at each node depends linearly on V even when reasoning in a restricted context. This is an artifact of the design of TABLEAU (and the context-building mechanism) that could be removed with some recoding. If this artifact were removed, run time for reasoning within the limited contexts would presumably not increase appreciably with problem size. In any case, run time in the contexts increases more slowly than run time for consistency checking in the entire theory. Accuracy (the percentage of correct answers from the consistency check) also seems relatively unaffected by problem size. We conjecture that, for large problems, accuracy is a function of r , C/V and L/V .

Combining these effects, we conclude that the effectiveness of context-limited consistency checking increases with problem size. The size of a radius r context, and thus the complexity of the consistency check, is essentially unchanged as V increases, but the accuracy of the consistency check does not seem to fall. This attractive property is due to the fact that, at least for underconstrained, random theories, the average length of the inference chains that might lead us to conclude $\neg d$ depends on C/V rather than on V . If this same effect occurs in realistic KBs then context limitation should be quite effective for large problems.

Experiment 2: The second experiment measures the effect of changing L on the effectiveness of context limitation. We fixed V at 200 and C at 400. We varied L from 20 to 180 by 20, generating 100 theories and testing 10 defaults per theory. Results are shown in Figure 2.

The most interesting result is the accuracy, which generally falls to a minimum at around $L = 0.4V$, and increases on either side of this point. We believe that the rise in accuracy below $0.4V$ is due not to any real increase in the effectiveness of the context-limited consistency check; below this point more defaults are consistent, and context-limited checks only make mistakes

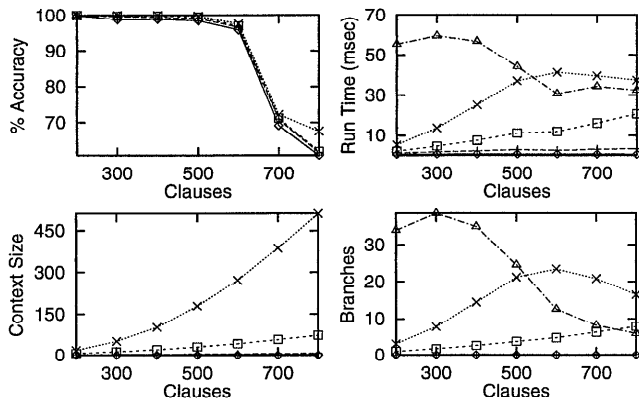


Figure 3: Accuracy, run time, context size, and search tree size, *vs* C , at $V = 200$, $L = 20$. $r = 0, 1, 2, 3$, and ∞ are marked \diamond , $+$, \square , \times and \triangle , respectively.

when defaults are inconsistent. However, note that as L falls, the difference in run time (and search tree size) between the context-limited check and the full check rises dramatically. Thus the results in Experiment 1 would have been even more favorable had we chosen a lower L/V ratio. Above $L = 0.5V$, the accuracy of the context-limited checks rises again. However, this region is not particularly interesting because so many literals are set by the input theory that full consistency checking becomes trivial.

Experiment 3: The final experiment measures how the effectiveness of context limitation changes with C . We fixed V at 200 and L at 20, and varied C from 200 to 800 by 100. This takes us from quite under-constrained to the edge of the transition region. We generated 100 theories and tested 10 defaults per theory, at each point. The results appear in Figure 3.

Here again *Accuracy* is the most interesting graph. Starting at about 600 clauses, or C/V about 3, the accuracy falls dramatically. We believe this is because near the transition region the interactions between the clauses in the theory become more global and any limited context is likely to miss some of them and so fail to detect inconsistencies. Our hope, of course, is that realistic theories of commonsense knowledge do not interact in this way (or do so only within local clusters that can be entirely included within the context).

One surprise is that starting at about 500 clauses, or a ratio of about 2.5, the cost of the consistency check in the radius 3 context rises *above* the cost of the full check. We believe this is due to of a kind of “edge effect” in the context. Consider a clause $x \vee y \vee z$ in the context. In some cases, there may be sets of clauses and literals in the full theory (e.g., $\neg x \vee a \vee b$, $\neg a$, and $\neg b$) but not in the context, that force the value of x . If this happens, the full check may actually be easier due to unit resolution. One way to test this hypothesis would be to unit resolve the input theory (this can be done in linear time) before any other reasoning is done.

Related Work

The idea of restricting the scope of consistency checking to a subset of the KB is not new. Our ideas are the logical result of a long tradition of context-limited AI reasoning systems dating back to CONNIVER (c.f. (McDermott & Sussman 1972; Fahlman 1979)). This line of work limits deductive effort, resulting in incompleteness. Limiting consistency checking in default reasoning, however, results in unsoundness—unwarranted conclusions may be reached due to lack of deliberation.

More directly related is Perlis’ suggestion to limit consistency checking to about seven formulae determined by immediate experience. Perlis argues that anything more is too expensive (Perlis 1984; Elgot-Drapkin, Miller, & Perlis 1987). He suggests that agents will have to simply adopt default conclusions and retract them later when further reasoning reveals contradictions. There are problems with Perlis’ approach, however. First, consistency-checking can be undecidable even in such tiny theories. More importantly, though, the errors this approach produces do not seem justifiable, since defaults are applied with essentially *no* reflection. Our analysis can be seen as explaining *why* (and when) such context-limited consistency checking can be expected to have a high probability of correctness. Furthermore, we believe that the notion of applying fast consistency tests in limited contexts provides significant leverage, allowing contexts to be larger while still achieving tractability.

THEORIST (Poole 1989) is also related in that it uses limited consistency checking to determine default applicability. However, THEORIST does not maintain a notion of context, so its errors are based on the use of incomplete reasoning mechanisms, rather than restricted focus of attention. Also, THEORIST has no notion of fast sufficient consistency checking.

Conclusions and Open Problems

We have described, and presented a preliminary experimental evaluation of, a practical way to trade accuracy for speed in consistency checking, that we expect to have applications to NMR (as well as to other commonsense reasoning problems that involve verifying consistency). We argue that restricting the consistency check to a focused context, combined with fast tests for consistency, can improve expected efficiency, at an acceptable and motivatable cost in accuracy.

The techniques we have outlined are not universally applicable—any gains from our approach hinge on the nature of the theories and defaults involved. It is easy to construct pathological theories in which *any* restriction of the context will falsely indicate consistency. In general, our approach will suffer if there are too many exceptions and those exceptions are hard to detect. We conjecture, however, that commonsense reasoning in general, and default reasoning in particular, is well behaved, in that complex interactions between distant parts of the KB are rare, and inconsistent defaults are

generally readily apparent. In addition, we achieve “asymptotic correctness”: if the agent has time to retrieve more formulae and reason with them, the probability of correctness increases. Thus, we can achieve a favorable trade of correctness for efficiency, without abandoning the semantic foundation provided by non-monotonic logic. Also, since by their very nature, defaults may be wrong despite being consistent with all one knows, agents should be prepared to accept errors in default conclusions, and deal with the resulting inconsistencies, as Perlis (1984) and many others have argued. An increase in the error rate should thus be less problematic to a default reasoner than it might be to a purely deductive system.

The efficacy of our approach depends on the design of both the context-generalization and consistency-checking mechanisms. These choices can only be based on, and ultimately verified by, extensive experiments with realistic commonsense KBs. Here we offered, and experimentally examined, only some simple first-cut mechanisms. In particular, our experiments use complete consistency checking in the context. In the propositional case, this appears to be sufficient; we believe that sufficient tests for consistency will be important primarily for first-order theories.

We reiterate that consistency checking is only one source of combinatorial complexity in default reasoning; for many problems of interest, default ordering presents another. We conjecture that limited contexts can also serve to limit the search of default orderings (e.g., by considering only defaults in the context), which may allow a tractable approximation of the overall default reasoning problem.

Acknowledgments

This work was supported in part by NSF grant IRI-94-12205, and ARPA/Rome Labs contracts F30602-93-C-0031 and F30602-95-1-0023. These ideas were refined over several years in discussions with many people, especially Alex Borgida and Bob Mercer. We are indebted to them and to Matt Ginsberg, Henry Kautz, David Poole, Bart Selman and anonymous referees, for fruitful discussions and criticisms. The remaining flaws are our responsibility, of course.

References

- Borgida, A., and Etherington, D. 1989. Hierarchical knowledge bases and tractable disjunction. In *Proc. KR'89*, 33–43.
- Brachman, R., and Schmolze, J. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science* 9(2):171–216.
- Cadoli, M., and Schaerf, M. 1992. Approximate inference in default logic and circumscription. In *Proc. 4th Int'l Workshop on Nonmonotonic Reasoning*.
- Cadoli, M.; Donini, F.; and Schaerf, M. 1994. Is intractability of non-monotonic reasoning a real drawback? In *Proc. AAAI-94*, 946–951.
- Cheeseman, P.; Kanefsky, B.; and Taylor, W. 1991. Where the really hard problems are. In *Proc. IJCAI-91*, 163–169.
- Crawford, J. M., and Auton, L. D. 1993. Experimental results on the crossover point in satisfiability problems. In *Proc. AAAI-93*, 21–27.
- Crawford, J. 1990. *Access-Limited Logic—A Language for Knowledge Representation*. Ph.D. Dissertation, University of Texas at Austin, Austin, Texas.
- Elgot-Drapkin, J.; Miller, M.; and Perlis, D. 1987. Life on a desert island: Ongoing work on real-time reasoning. In *Proc. 1987 Workshop on the Frame Problem in AI*, 349–357.
- Fahlman, S. 1979. *NETL: a System for Representing and Using Real-World Knowledge*. MIT Press.
- Gogic, G.; Kautz, H.; Papadimitriou, C.; and Selman, B. 1995. The comparative linguistics of knowledge representation. In *Proc. IJCAI-95*, 862–869.
- Gottlob, G. 1992. Complexity results for nonmonotonic logics. In *Proc. 4th Int'l Workshop on Nonmonotonic Reasoning*.
- Grice, H. 1975. Logic and conversation. In Cole, P., and Morgan, J., eds., *Syntax and Semantics, Vol 3: Speech Acts*. Academic Press.
- Kautz, H., and Selman, B. 1989. Hard problems for simple default logics. In *Proc. KR'89*.
- Levesque, H. 1984. A logic for implicit and explicit belief. Technical Report FLAIR TR 32, Fairchild Laboratory for Artificial Intelligence Research.
- McDermott, D., and Sussman, G. 1972. The CONNIVER reference manual. Technical Report AI Memo 259, MIT Artificial Intelligence Laboratory.
- Mitchell, D.; Selman, B.; and Levesque, H. 1992. Hard and easy distributions of SAT problems. In *Proc. AAAI-92*, 459–465.
- Perlis, D. 1984. Nonmonotonicity and real-time reasoning. In *Proc. 1st Int'l Workshop on Nonmonotonic Reasoning*.
- Poole, D. 1989. Explanation and prediction: An architecture for default reasoning. *Computational Intelligence* 5(2):97–110.
- Reiter, R. 1978. On reasoning by default. In *Proc. TINLAP-2*, 210–218.
- Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13:81–132.
- Selman, B., and Kautz, H. 1991. Knowledge compilation using Horn approximations. In *Proc. AAAI-91*, 904–909.
- Stillman, J. 1992. The complexity of propositional default logics. In *Proc. AAAI-92*, 794–799.
- Winograd, T. 1980. Extended inference modes in reasoning by computer systems. *Artificial Intelligence* 13:5–26.