

Toward Efficient Default Reasoning (Extended Abstract)

David W. Etherington James M. Crawford

AI Principles Research Department

600 Mountain Ave., P.O. Box 636

Murray Hill, NJ 07974-0636

{ether, jc}@research.att.com

Abstract

Early work on default reasoning was motivated by the need to formalize the notion of “jumping to conclusions”. Unfortunately, most existing theories of default reasoning require explicitly considering every possible exceptional case before applying a default rule. They are thus inherently undecidable in the first-order case, and remain intractable in all but the most restrictive cases. One possible approach to tractable default reasoning is to restrict consistency checks (which engender much of the intractability of default reasoning) to a restricted context. Unfortunately, consistency checking is undecidable in the first-order case, so a small context in no way guarantees tractability. Another idea is to use linear or polynomial time, but incomplete, consistency checks. Unfortunately, the known tractable checks generally fail in knowledge-bases of realistic size and complexity. We propose an approach to tractable default reasoning that combines the notions of limited contexts and fast checking, and argue that it overcomes the limitations of either alone. Our approach trades correctness for speed, but we argue that the nature of default reasoning makes this trade relatively inexpensive and intuitively plausible.

1 Computation and Default Reasoning

It is striking how much early work on nonmonotonic reasoning was motivated by the idea that defaults should make reasoning easier. For example, Reiter [1978] says “[the closed-world default] leads to a significant reduction in the complexity of both the representation and processing of knowledge”. Winograd [1980] observes that the nature of the world is such that mechanisms for making assumptions are necessary for an agent to act in real time: “A robot with common sense would

begin an automobile trip by walking to the place where it expects the car to be, rather than sitting immobilized, thinking about the infinite variety of ways in which circumstances may have conspired for it not to be there. From a formal point of view, there is no way to prove that it is still there, but nevertheless one must act, and must do so on the basis of plausible assumptions.”

Paradoxically, formal theories of nonmonotonic reasoning 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—it might take forever (not just a very long time) to determine that something is a consequence of a default theory. Even very restricted sublanguages based on propositional languages with linear decision procedures remain NP-complete [Kautz and Selman, 1989]. Convincing examples of useful theories within demonstrably tractable sublanguages for nonmonotonic reasoning have yet to appear.

Instead, we propose a framework for efficient default reasoning in which default conclusions can be drawn *without* having to consider every possible exceptional case. We show how very fast, incomplete, techniques can be used to test for exceptions. In general, these fast tests are unsuitable in knowledge-bases of realistic size and complexity. Despite this, we argue that fast incomplete consistency tests can be quite useful if one’s attention is restricted, using a very simple notion of relevance, to a limited context. The idea of limited search for exceptions thus synergizes with the fast testing for consistency to provide a framework for building feasible default reasoners.

Our hope is 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. However, this tradeoff is in accord 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.

This paper is organized as follows: section 2 reviews

the basics of Reiter’s Default Logic, section 3 discusses fast incomplete consistency tests, section 4 presents several approaches to context selection, and section 5 explains how these two ideas can be combined to achieve tractable default reasoning.

2 Default Logic

For expository purposes, we restrict our discussion to Reiter’s Default Logic [Reiter, 1980], although it is important to note that our ideas apply directly to other nonmonotonic formalisms. A *default* has the form:

$$\frac{Prereq(\bar{x}) : Just(\bar{x})}{Conseq(\bar{x})}, \quad (1)$$

where *Prereq*, *Just*, and *Conseq* 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 if *Prereq* is believed, and *Just* is consistent with what is believed, then *Conseq* should be believed: informally, this says that anything satisfying *Prereq* typically satisfies *Conseq* unless it is known not to satisfy *Just*.

3 Fast Sufficient Tests for Consistency

Nonmonotonic formalisms sanction default conclusions only if certain facts can be shown to be consistent with the rest of the system’s beliefs. For example, as explained above, in default logic the justification of a default must be shown to be consistent with all other beliefs before the consequent is sanctioned; similarly, in closed-world reasoning, $\neg P$ can be assumed only if P does not follow. Unfortunately, in the worst case, consistency is even harder to determine than logical consequence. While the theorems of a first-order theory are recursively enumerable, there can be no effective procedure to determine first-order consistency. In most cases, the unacceptable complexity of nonmonotonic formalisms can be traced to this consistency check.

Efficiency can thus be achieved in two ways: by skipping the consistency check, or by doing it quickly. Ginsberg [1991] proposes the former approach, suggesting that, in some applications (such as planning), defaults can be used as heuristics to guide the search for a solution. In such applications, consistency need not be checked when defaults are applied, since the final solution can be quickly verified. Further, the expected savings from default-guided search outweigh the costs of having to discard plans if this verification fails.

Ginsberg’s solution is not appropriate in many applications of default reasoning, however, because solution verification again entails consistency checking. This argues in favor of making the consistency check faster. Unfortunately this can only be done correctly in restricted sublanguages (which have yet to prove useful). There remains one potential avenue of escape: the intractability

of checking consistency is only a worst-case result. There are fast sufficient tests for consistency. A *sufficient consistency test* is one whose success implies consistency, but whose failure does not necessarily provide any information. The time complexity of some of these tests is no worse than linear in the size of the knowledge base.

For example, consider the default $\frac{\alpha : \beta}{\beta}$. In order to conclude β from a theory containing α , it must first be established that β is consistent with the theory. So long as the theory and β are each self-consistent, it suffices (but is not necessary) that none of the literals in $\neg\beta$ occur in the clausal representation of the theory. This can be determined in linear time (generally less if the theory is indexed), even for non-clausal theories.

Similarly, if the theory and β are *definite* (each clause contains exactly one positive literal), then β is consistent with the theory. Testing definiteness for a new β is independent of the size of the theory and depends only on β (provided that the definiteness of the theory is maintained as the theory evolves).

A slightly more complicated test derives from the techniques of [Borgida and Etherington, 1989], which can be used to quickly provide upper-bound checks that will succeed only if (but not necessarily if) consistency obtains (provided the system’s knowledge about the world is structured hierarchically). There are many other fast sufficient tests for consistency, but the foregoing suffice to illustrate our proposal for efficient default reasoning.

There is one obstacle to using fast consistency tests: in realistic applications such tests can be reasonably expected to fail. (This may explain why they have not heretofore been widely touted as eliminating intractability.) It would be a peculiar knowledge base that, for example, had the default “Birds usually fly”, $\frac{Bird(x) : Flies(x)}{Flies(x)}$, without *any* information about birds that don’t fly! The very representation of a rule as a default seems to presume knowledge (or at least the strong expectation) of exceptions. Hence, when the time comes to test the consistency of $Flies(Tweety)$, we can count on the presence of facts like $\forall x. \neg Penguin(x) \vee \neg Flies(x)$. This will cause the tests described above to fail, giving us no useful information. Notice, however, that in commonsense reasoning one is almost always completely unconscious of this or of the many other ways a bird might be shown not to fly. It seems plausible, then, that one might be able to restrict the system’s “focus of attention” to a subset of the knowledge base (a *context*) in such a way that it is likely that the fast consistency checks succeed.

4 Context-Limited Consistency Checking

We now turn to the problem of selecting the context relevant to checking the consistency of a default. Our

motivation here is two fold. First, at least in the propositional case, consistency checking is exponential in the size of the theory, T .¹ Clearly, if we need only check a subset, T' , for consistency, where $|T'| \ll |T|$, efficiency will improve significantly. Second, as we will show in section 5, one can use fast consistency checks and limited contexts together as a general approach to tractable default reasoning (even in first-order logic where full consistency checking is undecidable).

We first present two caricatures; obviously, neither should be taken seriously, but each provides useful insights into the limits of what might be achieved. We then propose one possible realistic mechanism for context selection. Finally, and most importantly, we argue that the nature of defaults makes the particular details of the context selection mechanism less critical than one might expect.

4.1 Implausible (But Illustrative) Context Selection

First, consider what we could hope for if we could be unreasonably successful in choosing a subset of the theory as the context. In particular, what if we should happen to choose the context for a consistency test of a formula, β , to be exactly those formulae in the theory that are relevant to a derivation of $\neg\beta$? Trivially, this would result in the fast sufficient test succeeding unless β is not consistent with the theory, in which case determining (in)consistency will take effort proportional to the difficulty of discovering the exception.

This result (if not the scenario) is intuitively appealing—providing the lowest complexity one could hope for, while preserving correctness. Whenever a default is applicable, it can be applied quickly, since exceptional classes are never present in the context unless they are relevant. It is thus unnecessary to explicitly consider and rule out every way an individual might be exceptional before concluding that it satisfies the default. On the other hand, if the default is contradicted, more computation may be required to detect this (as one would intuitively expect). In either case, correctness is preserved.

So, by a not-very-subtle sleight of hand, we have solved the problem of consistency checking by reducing it to the arguably harder problem of determining relevance. The point is merely that a well-chosen context can speed up default reasoning without necessarily incurring an unacceptable loss of precision. In section 4.3, we argue that it is not actually necessary to get the context exactly “right” to achieve significant performance improvements.

At the opposite extreme, consider choosing a very small context *at random* from a much larger knowledge base. If the size of the context is much smaller than the number of predicates in the language (given a knowledge

base language rich enough to express, say, an adult human’s knowledge about her world, such a context might still be quite large), then there is clearly a high probability that none of the literals of the formula being checked occur negatively in the context, and our sufficient test is likely to succeed.

Of course, the success of the sufficient test no longer assures us that the checked formula is consistent with the knowledge base, so the accuracy—while better than not checking at all—is little better than random. However, we do get a high probability that the default can be applied quickly, with very little overhead in building the context.

4.2 A Plausible Definition of Context

Naturally, any realistic context-selection mechanism will fall between these two extremes. Practical context selection seems to necessarily involve a balance between the cost of determining the context and the loss of accuracy from limiting consistency checking to the context. We now sketch one possible realistic context-selection mechanism (among many).

Consider an agent with a large knowledge-base of ground facts and quantified formulae. The notion of context we have in mind corresponds to the common-sense idea of her current focus of attention, and is partly determined by her recent history. In particular, facts come into the context as they are attended to (*e.g.*, from perception or memory), and exit as they become stale (or are squeezed out). The context should include any ground facts known about the objects under discussion (*e.g.*, *Tweety*). Further, the context should contain any rule whose antecedent and consequent are instantiated in either the context or the negation of the justification to be checked (*e.g.*, if *Penguin(Tweety)* is in the context, and the agent is checking the consistency of *Flies(Tweety)*, this should draw in $\forall x. Penguin(x) \supset \neg Flies(x)$). This amounts to a fairly standard skeleton for an agent’s context (*c.f.* [Elgot-Drapkin *et al.*, 1987]). With good hash coding and indexing, it should be possible to compute such a context quickly (one could also use a parallel algorithm such as spreading activation).

This simple notion of context can be elaborated in many ways. Limited forms of rule chaining can be provided in cases where chaining can be tightly bounded by the structure of the represented knowledge. For example, if the KB has a terminological component (*e.g.*, KL-ONE [Brachman and Schmolze, 1985] and similar languages), chains through the terminological hierarchy might be brought into the context by treating any deduction from terminological knowledge as a single ‘rule’ application. One could also retrieve “obvious” related items for the context using Crawford’s notion of the accessible portion of the knowledge-base [Crawford, 1990], Levesque’s notion of limited inference [Levesque, 1984], or other mechanisms that guarantee small computational costs for re-

¹Assuming $P \neq NP$.

trieval. In a longer version of this paper [Etherington and Crawford, 1992], we discuss the context-selection process in more detail.

4.3 The Mitigating Nature of Defaults

We now argue that such a compromise between the extremes of context-selection mechanisms can achieve a favourable balance between expected computational cost and expected accuracy. Clearly context-selection is a hard problem, and here we have described only a very rudimentary mechanism to do it. Fortunately, the nature of defaults makes selection of a useful context less difficult than might be expected, and gives our approach a high expected utility.

For a default to be reasonable, we contend, (at least) two factors must combine favorably. These are: (1) the likelihood that the consequent holds given that the prerequisite holds and, (2) the likelihood that if the prerequisite holds but the justifications are not consistent (causing the default to be inapplicable), then the agent will be aware of this fact. If the default is extremely likely to apply, one can tolerate the fact that one may overlook the odd exception. Similarly, if exceptions are easy to spot, it may be useful to have a default that rarely applies (corporate policy defaults come to mind). However, if exceptions are common but difficult to detect, one is ill-advised to make assumptions.² We develop this argument more fully elsewhere [Etherington and Crawford, 1992].

Another mitigating factor is the way that defaults are used. For example, when using defaults with respect to communicated information, Gricean principles of cooperative conversation [Grice, 1975] seem to enforce the second property: if the speaker believes that the hearer may draw an inappropriate default conclusion from her utterance, she is obliged to explicitly prevent it. This amounts to making sure that the appropriate contradiction will be in the hearer’s “awareness set” (context) when the default is considered. Similarly, in many non-conversational applications of default reasoning, exceptions are perceptible, and hence can be expected to be in the current focus of attention if they occur. For example, when planning to take one’s default (but flood-prone) route home, one can easily see whether it is raining heavily and, if so, block the default.

Finally, since by their very nature, defaults may be wrong despite being consistent with all one knows, it should be the case that agents would be prepared to accept errors in default conclusions. An increase in the error rate should thus be less problematic than in deductive reasoning.

Now observe that if we characterize a “good default” as one for which the probability that the prerequisite holds and the justification is inconsistent, combined with

²For now, we ignore the obvious third factor: the cost of being wrong.

the probability that an inconsistency will not be noticed, is low, we are guaranteed that a context-based system will produce results as good as its defaults. If an application requires defaults for which ensuring high expected correctness requires an unmanageable context, the gains from our approach will be less significant.

4.4 Examples

Consider the canonical default reasoning example:

$$\forall x. \text{Canary}(x) \supset \text{Bird}(x), \quad (2)$$

$$\forall x. \text{Penguin}(x) \supset \text{Bird}(x),$$

$$\forall x. \text{Penguin}(x) \supset \neg \text{Flies}(x), \quad (3)$$

$$\forall x. \text{Ostrich}(x) \supset \text{Bird}(x),$$

$$\forall x. \text{Ostrich}(x) \supset \neg \text{Flies}(x),$$

$$\forall x. \text{Emu}(x) \supset \text{Bird}(x),$$

$$\forall x. \text{Emu}(x) \supset \neg \text{Flies}(x), \quad \dots$$

Canary(Tweety),

Penguin(Opus),

Emu(Edna), ...

$$\frac{\text{Bird}(x) : \text{Flies}(x)}{\text{Flies}(x)}$$

where the ellipses indicate axioms about many other kinds of birds and many other individual birds. In standard default logic, to conjecture that Tweety flies one must show that *Flies(Tweety)* is consistent with the above theory—*i.e.*, that one cannot prove *Penguin(Tweety)*, *Ostrich(Tweety)*, etc. This amounts to explicitly considering all the ways that Tweety might be exceptional, which seems directly opposed to the way people use defaults.

On the other hand, provided that recent history hasn’t brought exceptional types of birds to mind, it is likely that our context would contain only *Canary(Tweety)* and (2). The sufficient test for consistency of *Flies(Tweety)* would then succeed, and so *Flies(Tweety)* could be assumed.

Considering whether Opus can fly, however, will bring *Penguin(Opus)* into the context and hence (3) (see section 4.2), resulting in the failure of the consistency test. Similarly, if one had been having a long discussion about various forms of flightless birds and then was asked whether Tweety could fly, one would expect facts about exceptional classes of birds to still be in the context. The fast test would thus fail, and one would have to explicitly rule out those types of exceptions.

4.5 Pathological Examples

Any gains from our approach hinge on the nature of the default theories involved. It is easy to construct pathological theories in which *any* restriction of the context will falsely indicate consistency. For example:

$$\forall x. Q(x) \vee P(1) \vee \dots \vee P(n)$$

$$\neg P(1) \vee \neg P(2) \vee \neg P(3) \vee \dots \vee \neg P(n)$$

$$\begin{array}{c}
P(1) \vee \neg P(2) \vee P(3) \vee \dots \vee P(n) \\
\quad \dots \\
\neg P(1) \vee \dots \vee \neg P(n) \\
\quad : \frac{\neg Q(x)}{\neg Q(x)}.
\end{array}$$

If any axiom is excluded from the context, our approach will infer $\neg Q(i)$, for some i , although this is inconsistent with the theory. Still, this is clearly a bizarre default theory, both since the default is universally inapplicable, and since the entire theory impinges on the applicability of a default. We believe that realistic default theories tend to be much less strongly-connected—not to mention much less frequently contradicted. It is clear, though, that our approach can only work if defaults are ‘reasonable’ (as discussed in section 4.3 above).

5 A Practicable Synergy

We propose a framework for efficient default reasoning based on combining limited contexts and fast consistency checking. The significant feature of our approach is the synergy between the two components: context restriction allows fast tests to be productive, and fast tests allow consistency to be checked in reasonable time. The combination of the two thus allows high expected (even absolute) efficiency despite the necessity for consistency checking.

Since it is an *approximation* technique, our approach is bound to introduce errors. Given the intractable nature of the problem, however, so is *any* efficient reasoner. Are the errors so induced reasonable? From the previous section it is clear that no categorical statement to this effect can be given for all theories. It is also clear that the quality of the results will ultimately depend on the appropriateness of the context-selection mechanism (or, equivalently, the reasonableness of the defaults).

This being said, we can make several arguments in favor of our approach. First, as argued above, the nature of the default reasoning task minimizes the impact of the loss of correctness while maximizing the gain in efficiency. Second, the kinds of errors induced by approximate consistency checking are intuitively plausible. People, lacking logical omniscience, frequently apply defaults that they “know” are inappropriate. For example, many of us can recall driving our default route between two points only to recall—on encountering a major traffic jam—having heard that the road would be under construction. Such examples seem to indicate that human default reasoning does take place in a limited context that does not include everything we know. As people temporarily “forget” about exceptions—making them inaccessible in the current context—so our approach will sometimes make inappropriate assumptions. Third, we achieve what has been described elsewhere [Borgida and Etherington, 1989] as “asymptotic correctness”, in the

sense that if the agent has time to retrieve more formulae and reason with them, the probability of correctness (measured, for example, against the specification of standard default logic) increases. Thus, we can achieve a favorable trade of correctness for efficiency, without abandoning the semantic foundation provided by non-monotonic logic.

The obvious danger implicit in our approach is that one will occasionally arrive at default conclusions that are inconsistent with what is already known. Any system based on these ideas must clearly be alert to the possibility of inconsistency, must be prepared to arbitrate among and retract contradictory facts, and must not allow an inconsistent knowledge base to license all conclusions. Elsewhere [Etherington and Crawford, 1992], we argue at length, as do Elgot-Drapkin *et al* [1987] and many others, that this is not a particular hardship for a default reasoner. In particular, the very nature of default reasoning entails that it will be possible for the system to make inferences that are consistent with all it knows, yet inconsistent with the state of the world. Such systems must be able to deal with the discovery of information that contradicts their default conclusions. Our approach admits the possibility that a contradiction may be detected purely as the result of further reasoning, rather than as the result of new observations; logically, the situation and the required responses are the same.

6 Related Work

The idea of restricting the scope of consistency checking to a subset of the knowledge base is not new. It seems to be a logical result of a long tradition of context-limited reasoning in AI systems dating back to CONNIVER (*c.f.* [McDermott and Sussman, 1972; Fahlman, 1979; McDermott, 1982]). This line of work limits deductive effort, resulting in incompleteness. Limiting the range of consistency checking in default reasoning, however, results in unsoundness—unwarranted conclusions may be reached due to lack of deliberation. However, this research should be a rich source of ideas on more sophisticated mechanisms for delineating contexts.

More directly related to the work presented here is Perlis’ suggestion to limit the check to a set of about 7 formulae determined by immediate experience. Perlis argues that anything more is too expensive [Perlis, 1984; Elgot-Drapkin *et al.*, 1987]. He suggests that agents will simply have to adopt default conclusions and retract them later when further reasoning reveals contradictions. 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 (thus improving accuracy) while still achieving tractability.

The THEORIST system [Poole *et al.*, 1987; 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.

7 Conclusions and Open Problems

We have outlined a practical approach to circumventing the unacceptable worst-case computational complexity of existing approaches to default reasoning. We argued that the restriction of consistency checking to the current context, combined with fast sufficient tests for consistency, yields a high expected efficiency, at an acceptable and motivatable cost in accuracy.

The techniques we outlined are not suitable for all defaults—as was shown in section 4.5. In general, our approach will suffer if there are too many exceptions and those exceptions are hard to detect. It seems, however, that the defaults people use specifically avoid these properties, since they could be expected to result in defaults of low utility.

Ultimately, the choice of a context-selection algorithm and of a set of fast consistency checks can only be based on experimentation. Given the defaults used in common-sense reasoning, one must determine empirically what sort of context allows most exceptions to be detected, and which consistency checks are sufficiently fast and powerful. One obvious next step is to implement a system along the lines described here, and study a number of examples. This will allow us to see whether the necessary cues are indeed available to make appropriate exceptions apparent to the default reasoner, and whether our notion of sufficient tests is fruitful.

We believe our arguments give convincing indications that context-limited reasoning can circumvent the inherent complexity of “correct” nonmonotonic reasoning. It is impossible to *prove* this, however, without reference to the particular knowledge and reasoning tasks presented to the system. One can always construct examples that require access to the entire knowledge base, on which any context-limited system will fail. Whether such perverse examples are common enough to constitute a significant impediment to this approach remains to be seen, but it seems unlikely.

Acknowledgments

This work arose from discussions over several years with many people, most especially Alex Borgida and Robert Mercer, who contributed heavily to the ideas presented here. We are indebted to them—and to Peter van Beek, Randy Goebel, David Poole, and Bart Selman—for many fruitful discussions, criticisms, and ideas. The remaining shortcomings remain solely our responsibility, of course.

References

- [Borgida and Etherington, 1989] Borgida, A. and Etherington, D. W. 1989. Hierarchical knowledge bases and tractable disjunction. In *Proceedings, First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Canada. Morgan Kaufmann. 33–43.
- [Brachman and Schmolze, 1985] Brachman, R.J. and Schmolze, J. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science* 9(2):171–216.
- [Crawford, 1990] Crawford, J.M. 1990. *Access-Limited Logic—A Language for Knowledge Representation*. Ph.D. Dissertation, University of Texas at Austin, Austin, Texas.
- [Elgot-Drapkin *et al.*, 1987] Elgot-Drapkin, J.; Miller, M.; and Perlis, D. 1987. Life on a desert island: Ongoing work on real-time reasoning. In *Proceedings of the 1987 Workshop on the Frame Problem in Artificial Intelligence*, Lawrence, KS. 349–357.
- [Etherington and Crawford, 1992] Etherington, D.W. and Crawford, J.M. 1992. On contexts and tractability in default logic. In preparation.
- [Fahlman, 1979] Fahlman, S.E. 1979. *NETL: a System for Representing and Using Real-World Knowledge*. MIT Press.
- [Ginsberg, 1991] Ginsberg, M.L. 1991. The computational value of nonmonotonic reasoning. In *Proceedings, Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA. Morgan Kaufmann.
- [Grice, 1975] Grice, H. P. 1975. Logic and conversation. In Cole, P. and Morgan, J. L., editors 1975, *Syntax and Semantics, Vol 3: Speech Acts*. Academic Press.
- [Kautz and Selman, 1989] Kautz, H. and Selman, B. 1989. Hard problems for simple default logics. In *Proceedings, First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Canada. Morgan Kaufmann.
- [Levesque, 1984] Levesque, H.J. 1984. A logic for implicit and explicit belief. Technical Report FLAIR TR 32, Fairchild Laboratory for Artificial Intelligence Research.
- [McDermott and Sussman, 1972] McDermott, D.V. and Sussman, G.J. 1972. The CONNIVER reference manual. Technical Report AI Memo 259, MIT Artificial Intelligence Laboratory.
- [McDermott, 1982] McDermott, D. 1982. Contexts and data-dependencies: A synthesis. Technical report, Department of Computer Science, Yale University.
- [Perlis, 1984] Perlis, D. 1984. Nonmonotonicity and real-time reasoning. In *Proceedings of the First International Workshop on Nonmonotonic Reasoning*, New Paltz, NY.

- [Poole *et al.*, 1987] Poole, D.; Goebel, R.G.; and Aleliunas, R. 1987. Theorist: a logical reasoning system for defaults and diagnosis. In Cercone, N. and McCalla, G., editors 1987, *The Knowledge Frontier: essays in the representation of knowledge*, New York. Springer Verlag. 332–352.
- [Poole, 1989] Poole, D. 1989. Explanation and prediction: An architecture for default reasoning. *Computational Intelligence* 5(2):97–110.
- [Reiter, 1978] Reiter, R. 1978. On reasoning by default. In *Proceedings of the Second Conference on Theoretical Issues in Natural Language Processing*, Urbana, IL. 210–218.
- [Reiter, 1980] Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13:81–132.
- [Winograd, 1980] Winograd, T. 1980. Extended inference modes in reasoning by computer systems. *Artificial Intelligence* 13:5–26.