Technical Report TIMLG-38, Turing Institute, Glasgow, UK (June 1990)
http://www.cs.utexas.edu/users/pclark/papers

Nonmonotonic Reasoning, Argumentation and Machine Learning

Peter Clark
The Turing Institute
36 N.Hanover St
Glasgow, UK

29th June 1990

Abstract

Machine learning and nonmonotonic reasoning are closely related, both concerned with making plausible as well as certain inferences based on available data. In this document a brief overview of different approaches to nonmonotonic reasoning is presented, and it is shown how the concept of argumentation systems arises. The relationship with machine learning work is also discussed. The document aims to highlight the links between nonmonotonic reasoning, argumentation and machine learning and as a result propose some potentially useful directions for new research.

1 Introduction

While first-order logic (FOL) has proved a powerful knowledge representation formalism, it has several limitations for AI applications. The two most severe of these are its **monotonicity** and its limited ability to represent **uncertain information**.

1.1 Monotonicity

The monotonicity of FOL stems from the use of a sound inference procedure, only inferring conclusions logically implied by a theory. The reason this is limiting is that, in real-world problems, we often wish to derive information about statements *not* logically implied by what we already know. For any statement `S` not logically implied by our theory `Δ`, first-order logic (FOL) tells us nothing. FO theories thus only provide **partial knowledge** of the world.

1.2 Representing Uncertainty

1.2.1 Types of Uncertainty

Much knowledge about the world is uncertain: however, we can often make statements *relating* to the truth of some proposition without actually knowing if that proposition is true or not. There are a variety of types of uncertainty, including:

Probabilistic: X is probably a Y.
Conditionals: if X then Y.
Disjuncts: X or Y.
Conditional evidence: if X then probably Y.

Relative Evidence: given X, Y is more likely.

Assumptions: Assume Y unless there's evidence to the contrary.

Determinations: Given X, Y can be determined.

Some of these are difficult or impossible to represent in FOL, limiting its utility. Some of the formalisms for nonmonotonic logic are more able to represent these kinds of uncertainty.

1.2.2 Probabilistic and Logical Uncertainty

Before discussing formalisms for nonmonotonic logic, a distinction between statements of **probability** and statements of **logic** should be made. These distinct statement types are frequently confused in the literature, leading to counterintuitive results. This distinction also allows nonmonotonic logics to be viewed as formalisms for handling a *particular type* of uncertainty.

Probabilistic statements ('most birds fly', 'typically birds fly' etc.) refer to statistics over a given population. Logic statements, however, refer to conclusions which, when drawn, are treated as 'definite' until further evidence comes along. Logic statements make no comment on the statistics about a given population.

A particular source of confusion is the phrase 'by default' (eg. 'by default birds fly') which colloquially can be used in both the probabilistic sense (eg. meaning 'most birds fly') and the logical sense (eg. meaning 'assume birds fly unless shown otherwise'). Although sometimes both interpretations are simultaneously valid, this is not necessarily the case.

The reason this distinction is important is because the valid axioms for inference differ for logical and probabilistic statements. For example, a version of transitivity can be used with a defaults ('Given A assume B unless B inconsistent; given B assume C unless C inconsistent; therefore given A, assume C unless B and C inconsistent') whereas no equivalent exists under a probabilistic interpretation of '$\Rightarrow$' ('most penguins are birds, most birds fly but *not* therefore most penguins fly'). Another example is the logical axiom

$$\mathtt{A} \Rightarrow \mathtt{B} \text{ implies } \mathtt{A} \wedge \mathtt{C} \Rightarrow \mathtt{B}$$

valid as an axiom for defaults but which is invalid when '$\Rightarrow$' is taken to mean 'are mostly' (In fact, $A \Rightarrow B$ and $A \wedge C \Rightarrow \neg B$ are not inconsistent under the probabilistic interpretation).

Probabilistic statements ('most birds fly') are often used in reasoning, but not all knowledge can be represented in this way. Logical statements are also often used, especially for conveying conventions ('assume normal conditions unless told otherwise'). As an example of this, McCarthy [14] points out that in the missionary-and-cannibals problem we assume the boat is not leaky on the basis of this logical convention rather than a probabilistic analysis of boat leakage likelihood. An ideal system for plausible reasoning should be able to represent both probabilistic and logical statements.

We can thus now define nonmonotonic reasoning as a particular kind of plausible reasoning, representing plausibility in this logical way (eg. defaults) rather than a probabilistic way. We will discuss ways in which the two can be combined later.

In the rest of this paper, we use the term 'default' in the logical rather than probabilistic sense. Buntine has formulated a logic for 'default' reasoning using a probabilistic interpretation of 'by default' to mean 'most' [2].

1.3 Their Relationship

Monotonicity and uncertainty representation are related (eg. it is not possible to draw tentative conclusions without being able to at least express them). However, the two should not be mixed. As we describe below, is possible to both create non-monotonic systems without allowing new types of uncertainty to be represented (eg. making a closed-world assumption), or alternatively to represent other forms of uncertainty while retaining monotonicity of inference (eg. Bayesian reasoning).

1.4 Argumentation Systems

If incorrect conclusions can be drawn then conflicts may arise, requiring **conflict resolution** knowledge to be included in the system. Addressing this problem leads to the notion of **argumentation systems**, where research has focused on the localisation and resolution of conflicting information.

1.5 Terminology

Monotonic Given a theory Δ, from which the sentences $T(\Delta)$ are derivable using our inference procedure, then adding sentences S to Δ can only make $T(\Delta \cup S) \supseteq T(\Delta)$.

Defeasible Synonym for non-monotonic.

2 Theory Completion Methods

A FO theory generally provides only partial knowledge of the world; it is silent on statements which can be neither proved nor disproved. A simple way of overcoming this is to make the **closed-world assumption** (CWA), that any statement is false if it cannot be proved true. The CWA augments a theory Δ with $\neg P$ for every P not logically implied by Δ. While simple, this method has several limitations:

1. It can produce inconsistent theories when there are disjuncts in Δ. For example, if Δ contains only
$$P(A) \vee P(B) \tag{1}$$
then both $\neg P(A)$ and $\neg P(B)$ should be added to Δ according to the CWA, but this is then inconsistent with equation 1. To avoid this problem, restrictions on the form of Δ can be imposed eg. only Horn clauses are allowed in Δ.

2. Given new information (ie. new sentences added to Δ), some of the assumptions made under the CWA, and their consequences, need to be retracted. Some method for keeping track of the dependencies of inferred sentences is needed to control this process.

3. The CWA only represents a single, specialised assumption. We'd like to also represent other assumptions (eg. 'assume apples are green').

A similar method called **predicate completion** (PC) can alternatively be used to make a CW-like assumption. Predicate completion augments a theory with formulae rather than negated literals. To complete for a predicate P, all clauses mentioning P are grouped and re-expressed in the form

$$\forall x\ E_1 \vee \ldots \vee E_k \Rightarrow P(x) \tag{2}$$

The completion formula to add is the same formula but with $\Rightarrow$ replaced by $\Leftarrow$, so the two together are equivalent to

$$\forall \mathtt{x}\ \mathtt{E}_1 \vee \ldots \vee \mathtt{E}_k \Leftrightarrow \mathtt{P(x)}$$

Thus, statement 2 of the form 'P(x) if Es' becomes 'P(x) if and *only* if Es', hence declaring P(x) to be false unless explicitly proved true by the Es. PC still suffers from the three limitations described above, although some disjuncts can be handled using 'predicate ordering' techniques [9].

Circumscription can be viewed as a more general form of these methods, and will cope with disjunctive formulae in Δ such as equation 1 above. Circumscription is applied to a particular predicate (P, say) in Δ. It operates by augmenting Δ with a formula (the circumscription formula) which states (in logic) that P(x) should be assumed false for all values of x except those for which P(x) *must* be true to maintain consistency with Δ. Given Δ contains only $\mathtt{P(A)} \vee \mathtt{P(B)}$, then the circumscription formula for P in Δ will not allow the truth of P(A) and P(B) to be derived but *will* imply that

$$(\forall \mathtt{x}\ \mathtt{P(x)} \Leftrightarrow \mathtt{x} = \mathtt{A}) \vee (\forall \mathtt{x}\ \mathtt{P(x)} \Leftrightarrow \mathtt{x} = \mathtt{B})$$

Again, the computation of the circumscription formula, and its use in inference, can be complex. Some simpler formulations of circumscription have been devised to ease this problem.

3 Representing Defaults

Theory completion methods allow us to make global assumptions about statements not inferred by our theory. However, we would also like to represent other more specialised assumptions about the world, such as defaults ('assume X by default'). Assumptions pervade much of everyday reasoning and hence are important to represent.

An ingenious method exists by which normal FOL can be combined with predicate completion (Section 2) to allow defaults to be expressed. This involves the introduction of **abnormality predicates**. We might write

$$\forall \mathtt{x}\ \mathtt{Bird(x)} \wedge \neg \mathtt{Abnorm1(x)} \Rightarrow \mathtt{Flies(x)}$$
$$\forall \mathtt{x}\ \mathtt{Ostrich(x)} \Rightarrow \mathtt{Abnorm1(x)}$$

expressing that 'all normal birds fly'. A theory completion method such as predicate completion is needed to prove ¬Abnorm(Tweety) given a normal bird Tweety, and hence derive Flies(Tweety). But if we are later told Tweety is an ostrich (adding Ostrich(Tweety) to our theory Δ), then Flies(Tweety) is no longer provable.

As this method requires predicate completion it still suffers from all the associated problems described in Section 2. It also restricts 'contrary evidence' to be specific features which we can enumerate. We cannot express, for example, that if making the default conclusion make the theory inconsistent then the default rule shouldn't apply.

To overcome this, and allow inconsistency checking as a method for blocking the use of default rules, several nonmonotonic logics have been proposed including:

1. McDermott and Doyle's **nonmonotonic logic** [14] introduces a modal operator M meaning 'is consistent', for example:

   $$\forall \mathtt{x}\ \mathtt{Bird(x)} \wedge \mathtt{M}\ \mathtt{Flies(x)} \rightarrow \mathtt{Flies(x)}$$

   means 'if x is a bird, and it is consistent to believe x flies, then x flies'.

2. Reiter's **default logic** [13] uses default rules of the form

$$\alpha : M\beta / \omega$$

meaning 'if α is known, and β is consistent with what is known, then assume ω'. For example,

$$Bird(x) : Flies(x)/Flies(x)$$

says 'if x is a bird, and it is consistent to believe x flies, then x flies'. Unlike McDermott and Doyle's nonmonotonic logic, the default rules are not formulas in the theory itself.

This test for global consistency as a condition for accepting a conclusion is a more general way of expressing defaults. However two problems dominate. Firstly, the test for consistency of β is computationally complex (even impossible in some cases) as not just β but all its implications (perhaps including more conclusions by default) must also be checked. Secondly, the logic is underconstrained in that there may be many alternative, consistent sets of conclusions which can be drawn ('multiple extensions') and no way to choose between them. An illustrative example is the theory with the two rules:

Assume Quakers are pacifists.
Assume Republicans are not pacifists.

What do we conclude about Fred, who is both a Quaker and Republican? These logics do not allow the provision of extra information to specify which rules should dominate in the event of a conflict.

Finally, Pearl has proposed a system for default reasoning which uses knowledge of **causality** to make default assumptions [12]. Given some effect `E`, a given cause `C` is assumed true by default *unless* some *other* possible cause $\mathtt{C}'$ of `E` is already known to hold. This method suffers, like default logic, from being unable to select between alternative extensions: given two possible causes $\mathtt{C}_1$ and $\mathtt{C}_2$, the system could assume either one was true by default (eg. choose one at random).

4 Direct Argumentation Systems

One of the problems with these default logics is that of deciding among competing extensions of a theory. This problem of resolving conflict, where evidence exists both for and against a given statement, is fundamental to uncertain reasoning, and hence to nonmonotonic reasoning as nonmonotonicity involves drawing uncertain conclusions.

Resolving conflicts requires knowledge about relations *between* the default or uncertain rules (we shall refer to them henceforth as 'arguments') used by the system. Systems where arguments are not explicit objects of the formalism, such as Reiter's default logic or circumscription, prevents such reasoning about the arguments. Konolige refers to these systems as **indirect argumentation systems** due to this limitation on their capacity for reasoning about arguments.

In contrast, **direct argumentation systems** do include arguments as objects of the formalism, allowing information such as relative argument strength to be incorporated in a natural manner. We now present several direct argumentation systems, grouped according to their strategies for conflict resolution.

Finally we note that argumentation does not necessarily imply nonmonotonicity. A plausible inference argument is not a certain inference because information is missing. If that missing information is obtained, we can then make that inference knowing it is now monotonic. If it is not, we can still make the inference, but only tentatively, knowing the conclusion may later have to be retracted. Alternatively, we can adopt a conservative approach whereby we only make the inference when all the missing information has been provided. In this latter case, the reasoning will then be monotonic with the advantage of computational simplicity but the disadvantage of limiting what can be inferred.

4.1 Specificity for Conflict Resolution

A major thrust of work on direct argumentation systems has been to derive *general* principles for adjudicating among conflicting lines of argumentation. The most prominent method is that of taxonomic inheritance with exceptions (eg. [7, 1]). Here, the principle of specificity is used to resolve conflict: information about subclasses overrides information about more general classes in the case of conflict. If Tweety is a bird, assume he flies, but if Tweety is an ostrich then assume he does not. However, while intuitively appealing this method does not resolve all conflicts: We still can't represent how to decide whether Fred, our Republican Quaker, is a pacifist or not.

4.2 Domain-dependent Principles for Conflict Resolution

Konolige has critisised this effort to formulate domain-independent conflict resolution strategies. He states that

> "general domain-independent principles [for adjudicating among conflict] will be very weak, and that information from the semantics of the domain will be the most important way of deciding among competing arguments" [10] p381

In support of this, he proposes a formalism called ARGH[1] for describing events [10, 11]. Arguments are classed into two types. A set of inference rules are used to resolve conflict, making use of information about argument type.

We briefly summarise the system, showing how the Yale shooting problem is encoded.

- The formalism is propositional, without variables or quantifiers.
- It uses a situation calculus, where a chronologically ordered set of **situations** $\mathtt{s}_0, \ldots, \mathtt{s}_n$ are deemed to occur.
- **Properties** of the world are always qualified by the situation in which they hold, for example $\mathtt{alive@s}_0$ represents that Fred (the Yale victim) is alive in situation $\mathtt{s}_0$.
- Two types of relation link consecutive situations together:
  a. **Events**, eg. `shoot`.
  b. **Persistence**, describing the tendency of properties to remain unchanged between situations in the absence of events.

Arguments describe properties of the world change between situations. For example

$$\mathtt{loaded@s}_0 \rightarrow_{\mathtt{shoot}} \mathtt{dead@s}_1, \mathtt{unloaded@s}_1$$

expresses that if the gun is loaded in situation $\mathtt{s}_0$ and a shoot event occurs, then, in the absence of evidence to the contrary, the victim will be dead and the gun unloaded in $\mathtt{s}_1$.

Note the qualification 'in the absence of evidence to the contrary' to drawing a conclusion, specified by the semantics of arguments in ARGH. This is equivalent to the

[1] ARGumentation with Hypotheses

qualification 'it is consistent to believe P' in default logic, though checking consistency is easier here due to ARGH's simplified language (ie. propositional, and without logical implication).

Instead of focusing on the problems of consistency checking, Konolige has progressed to ask how conflicts should be resolved when there *are* arguments to the contrary. Two general principles are proposed which can be paraphrased:

Persistence Defeat Event arguments defeat persistence arguments, ie. consequences of actions over-ride the tendency of world properties to remain unchanged.

Known Fact Defeat A given initial fact defeats any arguments against that fact.

These principles start to address the conflict resolution problem, but again will not deal with all conflicts (we still can't represent how to decide whether our Republican Quaker is a pacifist). More significantly, the principles are not 'watertight' – Konolige himself points out that persistence defeat may not always be appropriate, and that it is the semantics of the events themselves which decide whether this is so. Consider the following example, while noting that properties have a tendency to persist backwards in time as well as forwards (ie. $\texttt{p@s}_{\texttt{i+1}} \rightarrow_{\texttt{persist}} \texttt{p@s}_{\texttt{i}}$). Given Fred is $\texttt{asleep@s}_0$, then woken up, but is again $\texttt{asleep@s}_2$, we would like to conclude Fred is $\texttt{awake@s}_1$ as the `woken_up` event defeats the backwards persistence of `asleep` from $\texttt{s}_2$ to $\texttt{s}_1$. However in the syntactically identical case where Fred is $\texttt{alive@s}_0$, then shot, but is still $\texttt{alive@s}_2$, we would like to conclude Fred is also $\texttt{alive@s}_1$ as the backwards persistence of `alive` is very strong. This example nicely illustrates the importance of domain-specific information for resolving conflicts, and also shows ARGH needs extending to allow the encoding and use of this knowledge.

4.3 Numeric Methods for Conflict Resolution

An alternative method for resolving conflicting arguments is to assign some numeric measure of argument 'strength' to each argument and use some calculus for combining strengths together to see whether 'pro' or 'con' arguments dominate. In Fox et als. system, conflicts are resolved by simply by counting the number of arguments 'for' and 'against', and assuming the majority wins [8].

This method suffers from the problem of assigning numeric strengths to arguments. A single number is a poor representation of the justification for why an argument is valid, how strong and reliable that justification is, and how it interacts with justifications for other arguments. We discuss this further in Section 5.

4.4 Precedence Methods for Conflict Resolution

Another technique for resolving conflicts is to exploit **previous precedents** where similar conflicts have occurred and the outcome known. Some researchers in **case-based reasoning** has adopted this method. At first sight, probabilistic rather than logical methods might seem more appropriate as previous cases of conflict resolution often only add statistical evidence that the same outcome will apply in a new case. However, there is an important class of domains where the same resolution should *always* occur for a given conflict of arguments – those requiring consistent human judgement. While the domain of law is an obvious example, it should be noted that consistency of judgement plays a fundamental role in many decision-making tasks. We can say that the arguments should completely **determine** the outcome, a concept developed by Russell in his theory of determinations [4].

Use of precedents has the advantage of avoiding a representation of how argument strength is calculated and compared. Instead, only the *results* of previous conflict resolutions are needed, without requiring knowledge of how those results were reached. However, avoiding such a representation also restricts the scope for resolving conflicts – if no appropriate precedents are available, the system cannot solve the current problem.

4.5 Argument Space

To describe some example formalisms, we introduce the concept of an **argument space**. Argument space is akin to example space, each point in the space representing an example. In argument space, however, each axis represents an *argument* (for/against a conclusion of interest) which might apply for a given case, rather than a descriptive feature. The axis has only two values, '0' and '1', the higher value representing the stronger support in favour of the conclusion (ie. '1' means 'argument applies' for pro arguments and 'does not apply' for con arguments). A point thus represents a certain set of applicable arguments, but the resolution of any conflict that may exist is unknown. We only know relative information: moving one step further away from the origin adds an argument in favour or deletes one against.

However, for some special points the resolution of conflicts *is* known – these are the points representing precedent cases with known outcomes. These act as 'anchor points', mooring the combinations of 'pro' and 'con' arguments to fixed outcomes.

4.5.1 Some Systems

We now mention some systems in terms of argument space.

1. In their system **Hypo**, Rissland and Ashley [15] refer to the axes of argument space as 'dimensions' – some dimensions are continuous, where the value of some parameter is deemed to strengthen/weaken the argument. Given an example with known outcome, further hypothetical examples with the same outcome can be generated by stepping towards/away from the origin in argument space. Hypothetical cases play an important role in legal argumentation.

2. In Clark's argumentation system a similar technique is employed. A NewCase is represented by a point in argument space. If an 'anchor' OldCase is found at that point too, the conclusion for OldCase is deemed to also apply to NewCase. Additionally, if we examine other cases with stronger 'pro' arguments by repeatedly stepping away from the origin, but reach an OldCase where the 'con' arguments still dominate, then the 'con' arguments are also deemed to dominate for NewCase (vice versa for stepping towards the origin). However, if no appropriate anchor points can be reached in this manner, the system is unable to resolve the conflict.

3. Gardner's system for legal reasoning similarly searches for previous precedents where an identical pattern of argument occurred in order to resolve argument conflicts and ambiguities in a legal case [18].

All these systems are unable to adjudicate conflicts where appropriate precedents do not exist. Ideally, though, we would like to attempt some judgement based on the precedents which we do have available. The question here is, how do the differences between the OldCase and NewCase affect the assignment of OldCase's conclusion to NewCase?

1. Several authors have proposed using metrics to determine which OldCase the NewCase is 'nearest' to, and adopt that OldCase's conclusion for NewCase. Thus a gradient of argument strength is established between known points in argument space. An example is the system Persuader [16].

   The problem with this approach is finding an adequate metric for assessing similarity. Although most systems use ad hoc techniques, it is difficult to see how to avoid this.

2. To avoid the user being faced with providing ad hoc numbers, the classification and knowledge acquisition system **Protos** acquires a symbolic *explanation* from the user describing the relevant similarity between an OldCase and a NewCase. A numeric metric of similarity can be generated from this pseudo-English chain of inference, based on each inference's 'relation' and 'qualifier'. For example, matching attributes 'petrol' and 'fuel' by the inference "petrol usually used_as fuel" might be given a strength of 0.9 ('usually') $\times$ 0.8 ('used_as'). These matching strengths are of course crude measures, so Protos also includes a feedback mechanism where an incorrect classification will 'tweak' the matching strength to strengthen/weaken its matching behaviour appropriately.

5 Justifications, Meta-Knowledge and Machine Learning

While the described systems reason with arguments, they do not account for where the arguments arise from in the first place. This is a major limitation to these systems, because knowledge *about* the arguments – a type of meta-knowledge – is essential for proper resolution of conflicts. Without such knowledge the validity of the arguments themselves (as opposed to their conclusions) cannot be inspected ('*why* should `X` be an argument for `Y`?'). In everyday argumentation it is common to question the validity of arguments others put forward, and often a level of detail will be 'pushed' to 'sub'-argue about whether a particular argument is valid. Toulmin, in his proposed schemata for argument structures, calls this meta-knowledge the 'backing' [17] for an argument, and considers it an integral component of argumentation. Here we refer to it as the 'justification' for the argument. Many expert systems lack representation of the justification for their rules, limiting their explanation capabilities. Cohen has similarly called for the explicit representation and use of this meta-knowledge (termed 'endorsements') in probabilistic reasoning [3].

In the described systems, the user is assumed to provide all the arguments manually. In contrast, some machine learning (ML) systems derive plausible relations automatically from provided data. In the latter case, if these relations are used as arguments, then the justification for them *is* available in the form of the data provided. Thus, there is an important potential relationship between machine learning and argumentation – machine learning generates and can justify arguments, while argumentation will reason with them. However this relationship has not been hitherto exploited. One important point to note is that rule induction methods for ML generate probabilistic relations rather than logical relations (the difference discussed in Section 1.2.2) and hence probabilistic rather than logical argumentation methods are appropriate.

6 Conclusions

Nonmonotonic logics are concerned with a particular type of plausible reasoning, which we have termed logical uncertainty (Section 1.2.2). Nonmonotonic reasoning implies conflict may exist, which in turn requires conflict resolution strategies. This leads to the notion of argumentation systems, an important but still poorly explored area in artificial intelligence.

The systems we have described address problems of nonmonotonic reasoning and argumentation, but there are three major criticisms which apply to most of them:

1. They do not account for where the arguments arise from in the first place
2. They do not argue for the validity of the arguments themselves (as opposed to their conclusions)
3. The conflict resolution strategies are either 'incomplete' (will not resolve all conflicts) or ad hoc (using arbitary numerical metrics).

As discussed in Section 5, we consider the explicit representation of the justification or 'backing' for arguments to offer the most promising way to overcome these problems.

References

[1] I. Brachman. 'i lied about the trees', or, defaults and definitions in knowledge representation. *AI Magazine*, 6(3), 1985.

[2] W. Buntine. Modelling default and likelihood reasoning as probabilistic reasoning. *Annals of Mathematics and AI*, 1990. to appear.

[3] P. R. Cohen. *Heuristic reasoning about uncertainty : An AI approach*. Pitman, London, 1985.

[4] T. R. Davies and S. J. Russell. A logical approach to reasoning by analogy. In *IJCAI-87*, pages 264–270, 1987.

[5] J. de Kleer. Back to backtracking: Controlling the atms. In *AAAI-86*, pages 910–917, 1986.

[6] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[7] S. E. Fahlman. *NETL: A System for Representing and Using Real-World Knowledge*. MIT Press, Cambridge, Ma, 1977.

[8] J. Fox, A. J. Glowinski, M. J. O'Neil, and D. A. Clark. Decision making as a logical process. In B. Kelly and A. Rector, editors, *Research and Development in Expert Systems V*, pages 160–175. University Press, Cambridge, 1989.

[9] M. R. Genesereth and N. J. Nilsson. *Logical Foundations for Artificial Intelligence*. Kaufmann, Ca, 1987.

[10] K. Konolige. Defesible argumentation in reasoning about events. In Z. W. Ras and L. Saitta, editors, *Methods for Intelligent Systems, 3*, pages 380–390. Elsevier, NY, 1988.

[11] K. Konolige and M. E. Pollack. Ascribing plans to agents. In *IJCAI-89*, volume 2, pages 924–930, 1989.

[12] J. Pearl. Embracing causality in formal reasoning. In *AAAI-87*, pages 369–373, 1987.

[13] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132, 1980.

[14] R. Reiter. Nonmonotonic reasoning. In H. E. Shrobe, editor, *Exploring Artificial Intelligence*, pages 439–481. Kaufmann, Ca, 1988.

[15] E. L. Rissland and K. D. Ashley. Hypotheticals as heuristic device. In *AAAI-86*, pages 289–297, 1986.

[16] K. Sycara. Resolving goal conflicts via negotiation. In *AAAI-88*, pages 245–250, 1988.

[17] S. E. Toulmin. *The Uses of Argument*. University Press, Cambridge, UK, 1958.

[18] A. v.d.L. Gardner. The design of a legal analysis program. In *AAAI-83*, pages 114–118, 1983.

Appendix: Truth Maintenance Systems

As an appendix, we briefly mention truth maintenance systems (TMSs) as they are closely related to NMR. TMSs should not be thought of as a formalism for nonmonotonic reasoning, but as a computational technique for *control* of nonmonotonic inference. Nonmonotonic reasoning may involve the retraction of conclusions, and all their consequences, should the current beliefs lead to a contradiction. While chronological backtracking (eg. Prolog) is a simple technique for changing assumptions in the light of contradiction, it is also inefficient as inferences completely unrelated to those causing the contradiction may also be undone (and then later redone) during backtracking. Truth maintenance offers a computationally more efficient method by maintaining records of which beliefs depend on which, and only 'undoing' inferences on which the buggy conclusion was depended.

Doyle originally proposed a TMS as part of his Master's Thesis in 1980 [6]. The system records statements and their dependencies, each statement called a **node**. Some nodes are axioms, and some are assumptions dependent on other nodes being appropriately believed ('in') or not believed ('out'). The system forward-chains, trying to maintain a consistent set of beliefs. When a contradiction occurs, the assumptions $A_1, \ldots, A_n$ leading to the contradiction are collected, one is selected at random (A_i, say), and new nodes created to encode that, *if* all the evidence previously needed to believe $A_1, \ldots, A_n$ is appropriately in or out, then A_i should *not* be believed. Thus the current set of beliefs is changed to avoid the contradiction.

Doyle's system is rather unweildy, the problem rooted in the attempt to maintain one single set of consistent beliefs. A more parsimonious alternative is deKleer's ATMS (Assumption-based TMS), which simply labels statements with the various sets of assumptions they depend on; there is no notion of whether statements are believed or not. Unlike Doyle's system, these assumptions are the *initial* rather than intermediate assumptions that the statement depends on – in other words, the initial assumptions are propogated through each inference using a small set of rules for combining assumption sets together. Rather than maintaining a single set of beliefs, the ATMS works to locate and exclude *in*consistent sets. Initially it forward-chains. When a contradiction is reached, the ATMS simply notes that the assumptions leading to it are contradictory. No further inference requiring this set of assumptions is performed. Conclusions based solely on inconsistent assumption sets are not available for further inference. Again, though, the system lacks goal-directed control of inference. More recent developments of ATMS are directed towards remedying this [5].