Hybrid argumentation systems for structured news reports

Anthony Hunter
Department of Computer Science
University College London
Gower Street
London WC1E 6BT, UK
a.hunter@cs.ucl.ac.uk

December 10, 2001

Abstract

Numerous argumentation systems have been proposed in the literature. Yet there often appears to be a shortfall between proposed systems and possible applications. In other words, there seems to be a need for further development of proposals for argumentation systems before they can be used widely in decision-support or knowledge management. We believe that this shortfall can be bridged by taking a hybrid approach. Whilst formal foundations are vital, systems that incorporate some of the practical ideas found in some of the informal approaches may make the resulting hybrid systems more useful. In informal approaches, there is often an emphasis on using graphical notation with symbols that relate more closely to the real-world concepts to be modelled. There may also be the incorporation of an argument ontology oriented to the user domain. Furthermore, in informal approaches there can be greater consideration of how users interact with the models, such as allowing users to edit arguments and to weight influences on graphs representing arguments. In this paper, we discuss some of the features of argumentation, review some key formal argumentation systems, identify some of the strengths and weaknesses of these formal proposals, and finally consider some ways to develop formal proposals to give hybrid argumentation systems. To focus our discussions, we will consider some applications, in particular an application in analysing structured news reports.

1 Introduction

Argumentation normally involves identifying relevant assumptions and conclusions for a given problem being analysed. Furthermore, this often involves identifying conflicts, resulting in the need to look for pros and cons for particular conclusions. This may also involve chains of reasoning, where conclusions are used in the assumptions for deriving further conclusions. In other words, the problem may be decomposed recursively.

Argumentation is an important cognitive activity that draws on complex and uncertain knowledge for decision-making and problems solving. Key types of argumentation are:

Prediction of probable outcomes: This involves looking for positive and negative influences on the probability of some outcome. It may also involve uncertainty about which factors hold, and uncertainty about the relative significance of the influences. An example application may be in medical prognosis.

Identification of possible ramifications: This involves looking for possible outcomes of, for example, events, news or decisions. This is complicated by not just looking for the most likely outcomes. It can also involve looking for less probable but more significant outcomes. An example application may be in risk control in asset management. This may be viewed as a form of deduction where deduction is some rational process deriving inferences from a set of assumptions.

Identification of possible preconditions: This involves looking for possible preconditions of, for example, events, news or decisions. This is complicated by not just looking for the most likely preconditions as for example in criminal investigation or medical diagnosis. It can also involve looking for less probable but more significant preconditions as for example in aviation risk analysis. This may be viewed as a form of abduction where abduction is some rational process deriving some of the assumptions used in deriving an inference.

Classification according to some criteria: This involves using some set of criteria to decide on the classification of an instance. In this case, the most probable classification is chosen (e.g. medical diagnosis), the classification that can be given without any reasonable doubt (e.g. decision by jury in criminal law), or the classification that can be given on the balance of evidence (e.g. decision by jury in civil law). Further criteria include those delineated by voting theory.

Given the importance of argumentation in professional life, there is much potential in developing tools for supporting the task. Formal approaches to argumentation systems have been developed as a branch of formal knowledge representation and reasoning. In contrast, informal approaches to argumentation have been proposed in various subjects including management, social sciences, and philosophy. We believe that the starting position for developing tools for argumentation should be a formal approach. However, we also believe that there are some important ideas in the informal approaches that could be harnessed to produce hybrid systems more appropriate for applications.

An excellent introduction to the notion of argumentation is by Toulmin [Tou58], with developments reviewed by van Eemeren et al [vGK87], and an excellent introduction to the role of logic in argumentation is Fisher [Fis88].

In this paper, we review some of the features of argumentation systems, and then consider the application area of argumentation with structured news reports. The application will help focus our discussions. We review the nature of structured news reports, and consider some of the requirements of tools for argumentation with such news reports. In the subsequent section, we review five key approaches to formalising argumentation. These are defeasible logics, coalition systems, default logics, coherence (consistency-based) systems, and causal networks. After considering the pros and cons of each of these approaches with respect to the requirements for argumentation with structured news reports, we look at the possibilities for hybrid argumentation systems. We look at some systems that are logic-based but also incorporate graphical notation to facilitate user interaction, and then consider further key topics for developing hybrid argument systems including ontologies and scenario analysis.

2 Features of argumentation systems

In formal and informal approaches to argumentation, we see a number of features that we can draw out and analyse. The common feature in argument systems is that they incorporate some representation of individual arguments (pros and cons) and techniques for comparing conflicting arguments. To expand on this idea, features that appear in some or all argumentation systems include:

Capturing factors with influences Factors are the pieces of information being taken into consideration. This may include information such as assumptions, facts, definitions, rules, regulations, theories, statistical statements, and defaults. Any argumentation system needs to support the representation of the factors being taken into consideration, and of the influence these factors may exert over other factors. We may think of a factor with an influence being an argument. In a logic-based approach, factors are represented in a logical language. In many of these proposals, a factor is represented by an atomic proposition, or set of formulae, and influences include undercutting (contradicting an assumption used in another factor), and rebutting (contradicting a conclusion of another factor). In a probabilistic approach, a factor may be a random variable and an influence may be a causal effect on another factor.

Representing types of influence From a formal perspective, types of influence include logical or probabilistic influence. In some formal approaches, preference orderings over arguments are used to capture a relative difference in significance. However, in an informal setting a much wider range of types of influence may be considered such as authoritative (eg. if a senior clinician says patient X has disease Y, this is a strong argument for believing that X has disease Y) or legislative (eg if legislation states that the maximum speed on the road is 120kmph, this is a strong argument for not driving faster than 120kmph). Many other types of influence may be considered (see for example [Wal89, Sil96, Sil97, MP99]). One way of putting them into a formal perspective is that there is implicit knowledge that is assumed with these other types of influence. We believe that if this other knowledge is made explicit, either at the object-level or the meta-level, then we can return to a logical or probabilistic setting.

Significance of influences Some influences may be more influential than others. However, the nature of influence seems diverse. Types of orderings of influences include "more likely", "higher risk", and "greater authority". Influence can also be quantified such as by a probability value or possibility value.

Aggregation of influences If there are a number of arguments for and against a particular conclusion, an aggregation function determines whether the conclusion holds in the framework. Often these aggregation functions can be described as being binary since they just consider the existence of arguments for and against, and so they are not sensitive to the number of arguments for or against. In most logic-based approaches, aggregation is nested within the definition for the consequence relation. In a probabilistic approach, aggregation can be undertaken using a probability distribution.

Graphical representations Graphical representations are often the basis of informal approaches. They are often simpler and more lucid then logic-based or quantitative approaches. Though some logic-based and quantitative approaches also incorporate graphical representations. Using a graphical approach, nodes can represent factors, and arcs can represent influences. Labels on arcs can be used to give more detailed information about the nature or degree of influence.

The above discussion is meant to indicate some commonality in argumentation systems, and provide a framework for comparison, though it conflates some important differences between approaches. Further ways that we can compare proposals for argumentation systems include:

Simplicity vs expressibility Simple notation is easier to understand and to use but may lack ways of representing certain kinds of information adequately. In contrast, more expressive languages may be rich enough to give adequate representation of some kinds of information but may be too difficult to understand or use by non-technical users. Furthermore, more expressive languages may be too computationally expensive to use automated reasoning technology.

Representation vs reasoning When using argumentation techniques in decision-support, often the representation of the arguments is sufficient. In other words, users are just interested in a static picture of a set of arguments and their interactions. However, in some applications there is the need to support reasoning. First this gives a formal definition of allowed inferences and second it opens the opportunity for automated reasoning technology. Between the extreme of having only a representation and the extreme of having only automated reasoning is the possibility of tools for syntax checking and proof or model checking (where the user constructs the proof or model and the system checks the correctness).

Push argumentation vs pull argumentation Push argumentation is based on a knowledge engineer constructing a knowledgebase for a user (or set of users) with the assumption that the user has no direct control over the knowledgebase. The knowledge is captured in a formalisation that the user is unlikely to be familiar with, and the knowledgebase is treated as a blackbox. In this case, the results of argumentation are pushed onto the user by automated reasoning technology. In contrast, pull argumentation is based on the user(s) being largely responsible for the construction of the knowledgebase. Hence the knowledgebase is not a blackbox, and the formalism is familiar to the user. In this case, the results of argumentation are pulled by the user. The user constructs the arguments to draw conclusions, though there is the possibly of technology being used to check correctness.

In general, informal approaches seem to be more closely related to particular types of argumentation used in practice. Of particular relevance are techniques developed as decision-support tools in management. In informal argumentation systems, there are proposals for richer notions of warrant. This helps users more quickly and confidently model some problem, and may help others to understand the results of the analysis. If we want to enhance formal approaches to argumentation so that they are more useful in applications, then it is worthwhile examining more closely the nature of formal approaches and considering how they can be developed with ideas from informal approaches.

3 Argumentation with structured news reports

Here we briefly review the idea of structured news reports. We will be using them to motivate some of our discussions of various proposals for argumentation systems.

3.1 Structured news reports

Structured text is an idea implicit in a number of approaches to handling information such as news reports. An item of structured text is a set of semantic labels together with a word, phrase, sentence, null value, or a nested item of structured text, associated with each semantic label. As a simple example, a report on a corporate acquisition could use semantic labels such as “buyer”, “seller”, “acquisition”, “value”, and “date”. A structured news report is just a news report represented in structured text. An example is given in Figure 1.

Some news agencies store news reports as structured text. In addition, new technologies, such as information extraction and XML, will massively increase the amount of structured text available. The advent of structured text raises the need for tools for structured text analysis.

```
⟨bid-report⟩
    ⟨bid-date⟩ 30 May 2000 ⟨/bid-date⟩
    ⟨buyer⟩
        ⟨company⟩
            ⟨name⟩ France Telecom ⟨/name⟩
            ⟨capitalization⟩ 150 Billion Euros ⟨/capitalization⟩
            ⟨headquarters⟩ Paris, France ⟨/headquarters⟩
        ⟨/company⟩
    ⟨buyer⟩
    ⟨target⟩
        ⟨company⟩
            ⟨name⟩ Orange ⟨/name⟩
            ⟨headquarters⟩ Bristol, UK ⟨/headquarters⟩
        ⟨/company⟩
    ⟨target⟩
    ⟨bid-type⟩ agreed ⟨/bid-type⟩
    ⟨bid-value⟩ 40 Billion Euros ⟨/bid-value⟩
    ⟨report-info⟩
        ⟨source⟩ Orange website ⟨/source⟩
        ⟨URL⟩ www.orange.co.uk ⟨/URL⟩
        ⟨report-date⟩ 31 May 2000 ⟨/report-date⟩
    ⟨/report-info⟩
⟨/bid-report⟩
```

Figure 1: An example of a simple news report in the form of structured text using XML notation.

Here we are interested in using argumentation systems for reasoning with the information in structured news reports. For this we assume that news reports are represented as structured text (eg in XML or in semi-structured data [Abi97, Bun97] or in a template output from an information extraction system [CL96, Gri97]) where the text entries are individual words, numbers or very simple phrases. We represent each news report as a conjunction of classical logic literals. For details on the process and viability of this see [Hun00a, Hun00d].

Once we have news in the form of logical literals, we can use the literals in some form of logical reasoning. For example, given Figure 1, we may wish to examine the argument for and against the bid by France Telecom for Orange being successful. Since it is an agreed bid, then failure is most likely to come from the bid being rejected by the anti-trust authorities. This argument would be even stronger if there is evidence that the two companies have significant overlaps in their customer or product bases. Another possibility for the bid being rejected is if the bid price is too low, then the Orange shareholders might vote against it, or if the bid price is too high, then the France Telecom shareholders may rebel. So given this news report, there seems to be a lot of background knowledge available that can be used to draw inferences for pros and cons for this bid eventually being successful.

In an argumentation system, we need to capture factors and their associated influences. However, it depends on the application as to the need for types of influence, significance, aggregation, graphical representation, push vs pull argumentation, and so on. Features of the application that can affect the choice include the form and availability of the background knowledge, the technical skills of the user group, whether pull or push argumentation is being used, and the reliance the users put on the inferences.

3.2 Requirements for argumentation with news

We now need to clarify our requirements for argumentation with news reports by considering the following types of desirable functionality. We are not suggesting that every argumentation tool should support all this. Rather there is a variety of possible functions that may be supported in a given argumentation tool.

Ramification analysis People constantly need to absorb new information, and to consider the ramifications of it. For example, on receiving the news that the US Federal Reserve Bank has decreased interest rates by 1%, there are numerous ramifications that could follow. For example, the dollar could go down in value, and the European Central Bank could follow suit. Ramifications of interest depend on the users of the argumentation. This raises interesting questions for knowledge representation and reasoning. In ramification analysis we need to determine both the likely outcomes from events occurring and the less likely, but very significant outcomes, from events occurring. Often this would seem to call for push argumentation with the knowledgebase being developed by a knowledge engineer, and the user looking at the results of reasoning. Nevertheless, the user needs to have some understanding of the ramifications, and so there may be a need for abstraction of the arguments used (see also "Abstraction of knowledge" below). Though, a user may want to construct the arguments and counter-arguments for particular ramifications, and therefore undertake pull argumentation.

Precondition analysis People also need to determine how events have occurred. This involves determining the possible or probable preconditions. Returning to the example of the US Federal Bank, preconditions of interest may include the possibility that the US economy might be sliding into recession, or that the US dollar is climbing too steeply. They also need to consider how events may occur in the future, and again need to determine possible or probable preconditions. In the later case, this would often call for pull argumentation since the user is looking for controlling the occurrence of an event. This is explicit in some informal approaches to argumentation. The knowledgebase is developed by the user. The emphasis is on representation of arguments. Usually the automated reasoning is limited or non-existent. However, automated reasoning may be useful even in pull argumentation. For example, if a user has used abduction to identify an explanation for some items of news, then automated reasoning can be used to check that the abduced information does indeed imply the item of news. For push argumentation in precondition analysis, we also need to consider the automation of finding the assumptions. In other words, we may need automated abduction technology.

Scenario analysis In a set of arguments, there may be a number of factors represented that can directly or indirectly influence a particular factor that is of interest to the user. So the outcome of this factor is regarded as contingent on the value of the other factors. To analyse this contingency, a scenario is an assignment of values to some of the other factors. Given this assignment, the values can be propagated in order to determine the value of the factor of interest. Scenario analysis involves identifying extreme scenarios, such as best-case and worst-case scenarios, and average-case scenarios to see the effect on the outcomes. Scenario analysis has not really been explicitly explored in logic-based argumentation, but we believe that it is a functionality that is called for in news analysis and decision making.

Abstraction of knowledge Abstractions can help users understand something of the underlying formal arguments without getting overwhelmed by the complexities of the underlying formalisms, knowledge or reasoning processes. Abstractions may be obtained in a number of ways including: presenting only key arguments or knowledge to the user; using a graphical notation to present the key arguments to the user; and giving a simplified version of the knowledge to the user.

Linkage of sources There is a need to be able to link (perhaps in the sense of a hyperlink) news articles to inferences and decision making. This is so as to use news articles as justification or warrants for particular influences. This is influenced by suggestions of Toulmin ([Tou58]) as illustrated in Figure 2. As an example, consider a user working for a property investment company who wants to make a decision on whether to invest in a new speculative office project in Warsaw. The user may start by looking for relevant articles from online information sources. These may be from information vendors such as Reuters or Dow Jones, or from in-house sources. These reports may describe the state of the economy, the reliability of contractors, the demand for office supply, the likelihood of big multinationals moving into Warsaw, and so on. Once the relevant articles and reports have been identified, the user constructs a directed graph that indicates the key factors together with the relative influences. This is a form of pull argumentation, where the net result is a structuring of evidence in the form of news reports. There is also a need to use linkage in push argumentation, so that for example inferences can be explained to users in terms of the arguments and the news reports used.

Synthesis of news reports In Section 3.1, we introduced structured news reports as data structures containing facts. Maybe some facts are incorrect, erroneous, or uncertain. But at the level of logic, they are ground literals. This is how we view structured news reports in the rest of this paper. However, news is often synthesised into longer news reports or reviews (eg an article in *The Economist* or an editorial in a national newspaper). Longer reports can clarify by providing relationships between news reports and other knowledge. Implicit in any longer news report or review is that it is based on a set of arguments with some logical structure. These arguments are made by parts of the text providing facts and warrants for intermediate inferences and overall conclusions. By analysing the warrants, these arguments can be made explicit in the form of a directed labelled graph (see for example [Fis88]). The emphasis is on discussing a number of news reports (either implicitly or explicitly) together with contextual information and interpretation. The relevance to argumentation with structured news reports includes: a synthesis can be used as an explanation to users for certain knowledge used in argumentation; and a synthesis can be analysed to obtain knowledge for a knowledgebase for argumentation.

Generalisation of news reports Given a number of structured news reports, a useful goal is to identify general patterns. For example, consider a number of news reports about companies listed on the stock market that reveal a significant fall in sales. For each of these news reports, there is high a probability that there will be another news report for each of these companies that reveals a significant fall in the share price published around the same time as the first news report on fall in market share. Identifying such patterns, and therefore generalising on news reports, is a potentially important way of generating knowledge for argumentation systems.

To summarise our requirements, there is a need for formal argumentation approaches for argumentation with structured news reports. Furthermore, there are requirements for lucid abstractions of sets of arguments, for the ability to link arguments with supporting information, which may involve a richer more specific notation, and for support for ramification analysis, precondition analysis, and scenario analysis. All of these requirements may be required in the context of both push and pull argumentation.

Primarily, we see news reports as input data for knowledgebased systems. In this way, each news report is treated as a set of facts. However, we may also view a news report, or set of news reports as a precursor to knowledge. In this way, we may use news reports to generate knowledge via data mining, or motivate, justify or explain (by abstraction) general statements in the knowledgebase by association (by linkage).

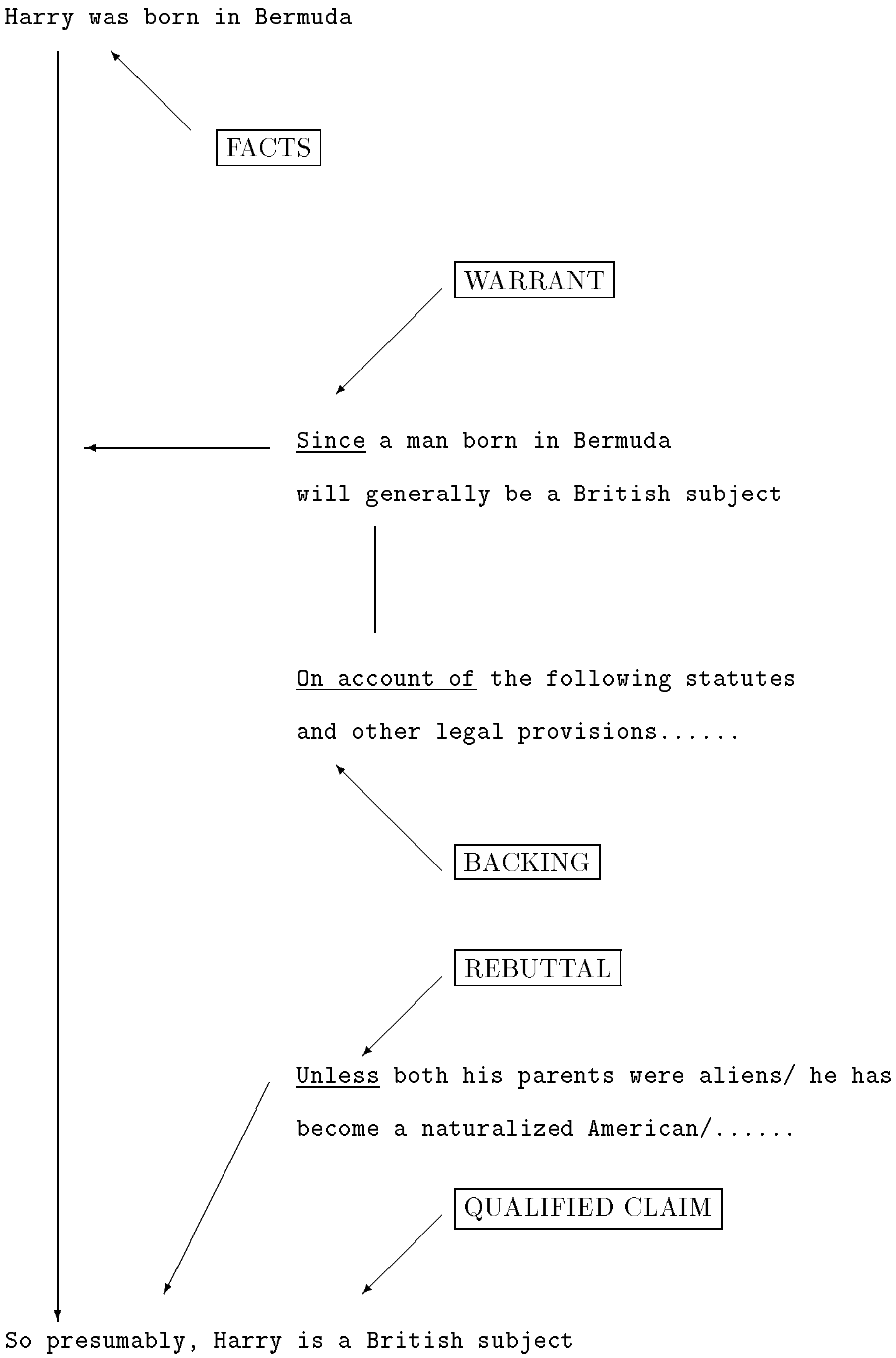

Figure 2: Example of argumentation taken from [Tou58].

In order to further delineate our concerns in this paper, we need to briefly review the monological and the dialectical views on argumentation.

Monological A single agent has the knowledge to construct arguments for and against a particular proposition. The knowledge may come from heterogeneous sources. After constructing the arguments, the agent may then draw some conclusion on the basis of the assembled arguments. The emphasis of the monological view is on how to construct the arguments and how to draw conclusions from the assembled arguments.

Dialectical A set of agents interact to construct arguments for and against a particular proposition. If an agent offers an argument, one or more of the other agents may dispute the argument. Agents may use strategies to persuade the other agents to draw some conclusion on the basis of the assembled arguments. The emphasis of the dialectical view is on the nature of the interactions, and on the process of building up the set of arguments until the agents collectively draw a conclusion.

We assume the focus of our interest is the monological view since we are trying to model the arguments that follow from news reports from the perspective of a single agent. We may consider the agent as being some kind of editor who is collating information prior to drawing some conclusion such as identifying some ramifications of interest. For push argumentation, we see the software system as being this editorial agent. For pull argumentation, we see the software system as being a tool used by this editorial agent.

4 Formal approaches to argumentation

There are many proposals for formalisms for logic-based argumentation. For general reviews of formalisms for argumentation see [FKEG93, GK98, VLG98, PV00, CML01]. Furthermore, some of these formalisms are being developed for applications in legal reasoning [Pra97], in medical reasoning and risk assessment [FD00], and in agent-based systems [PSJ98]. A review of argumentation systems that relate proposals to potential application areas in knowledge engineering, decision-support, multi-agent negotiation, and software engineering, is given in [CRL00].

In this section, we will consider key proposals for defeasible logics, coalition systems, default logics, coherence (consistency-based) systems, and causal networks in more detail, and in particular consider how they meet the requirements for argumentation with news. Our presentation will be biased towards building argumentation systems rather than the subtleties of the formalisms.

We have taken a broader interpretation of formalisms for argumentation than is often used in the argumentation logics literature. Our reasoning for taking a wider view is that as the role of argumentation is being developed, then there are formalisms developed for other roles, such as in non-monotonic reasoning, that are potentially useful in argumentation. However, we believe that there are further formalisms that could also be included, in particular paraconsistent logics such as C_ω [dC74], four-valued logic [Bel77], and quasi-classical logic [Hun00c, Hun01a].

4.1 Defeasible logics

There are a number of proposals for defeasible logics. The common feature for these logics is the incorporation of a defeasible implication into the language.

Defeasible logics have their origins in philosophy and were originally developed for reasoning problems similar to those addressed by non-monotonic logics in artificial intelligence. In [Pol87, Pol92], Pollock conceptualises the notions of reasons, prima facie reasons, defeaters, rebutting defeaters, and undercutting defeaters, in terms of formal logic. Arguments can then be defined as chains of reasons leading to a conclusion with consideration of potential defeaters at each step. Different types of argument occur depending on the nature of the reasons and defeaters. This has provided a starting point for a number of proposals for logic-based argumentation.

The defeasible argument system, of [SL92], extends a classical knowledgebase with defeasible (meta-level) rules of the form $\alpha \Rightarrow \beta$, where if α is inferable, then β is inferable. Each rule may have free variables that can be universally instantiated.

From a knowledgebase of classical (non-defeasible) formulae, plus defeasible (meta-level) rules, argument "for" and "against" any particular inference can be constructed. Each argument is a proof tree, where each step in the proof tree is either an application of modus ponens or an application of universal instantiation. In case of conflict, "more informed" arguments are preferred, as illustrated by the following examples.

Example 4.1 *Consider $\{\alpha \wedge \beta \Rightarrow \gamma\}$ as an argument for γ using the non-defeasible set $\{\alpha, \beta\}$ and $\{\alpha \Rightarrow \neg\gamma\}$ as an argument for $\neg\gamma$ using the non-defeasible set $\{\alpha\}$. Here the non-defeasible set $\{\alpha, \beta\}$ can also be used to derive $\neg\gamma$ from the argument $\{\alpha \Rightarrow \neg\gamma\}$, but we cannot use the non-defeasible set $\{\alpha\}$ used with the second argument to derive γ using the rule in the first argument. In other words, $\{\alpha \wedge \beta \Rightarrow \gamma\}$ with $\{\alpha\}$ is not an argument for γ. So the first argument is more informed than the second argument.*

Example 4.2 *Consider $\{\alpha \Rightarrow \neg\gamma\}$ as an argument for $\neg\gamma$ using the non-defeasible set $\{\alpha\}$ and $\{\alpha \Rightarrow \beta, \beta \Rightarrow \gamma\}$ as an argument for γ using either of the non-defeasible sets $\{\alpha\}$ or $\{\beta\}$. Here the non-defeasible set $\{\alpha\}$ can also be used to derive $\neg\gamma$ from the rule $\{\alpha \Rightarrow \neg\gamma\}$ in the first argument, but we cannot use the non-defeasible set $\{\beta\}$ from the second argument to derive $\neg\gamma$ using the rule in the first argument. In other words, $\{\alpha \Rightarrow \neg\gamma\}$ with $\{\beta\}$ is not an argument for $\neg\gamma$. So the first argument is more informed than the second argument.*

Another approach that may be described as based on defeasible logics is that of abstract argument systems [Vre97]. An abstract argumentation system is a collection of "defeasible proofs", called arguments, together with an ordering relation that represents the relative difference in force of the arguments. The system does not assume a particular underlying language, and so no consideration is given to connectives including negation. Though the language does include $\perp$ as a symbol. Assuming $\phi_1, ..., \phi_{n-1}, \phi_n$ are object-level formulae, then strict rules of inference are of the form $\phi_1, ..., \phi_{n-1} \rightarrow \phi_n$ and defeasible rules of inference are of the form $\phi_1, ..., \phi_{n-1} \Rightarrow \phi_n$. So if there are arguments for the premises of a rule holding, then there is an argument for the consequent holding. In this way, rules can be composed into trees to give more complex trees.

The framework focuses on incompatibility of sets of arguments, as opposed to inconsistency of pairs of arguments, and defeat is defined in terms of incompatibility and the ordering relation over arguments so that in case of incompatibility, less preferred arguments are defeated. This is used via a notion of inductive warrant. We start with a fixed base set of arguments that is a finite compatible subset of the language, and then generate all arguments from this. This gives the level 1 arguments. At level 2, some arguments are defeated by arguments not defeated by arguments at level 1. At level 3, some arguments of level 2 that were defeaters at level 2, are defeated by arguments at level 2 and so the arguments they defeated are reinstated. This process continues at level 4,5,... until a fixpoint is reached.

In practice, and for comparison with other defeasible logics, we can treat the meta-level symbols $\rightarrow$

and $\Rightarrow$ as object-level symbols. The practical advantage of this is we then have a relatively simple language together with an intuitive strategy for deciding which arguments should ultimately hold. Some users may appreciate a language that does not incorporate the usual connectives and does not force incompatibility to equate with α and $\neg\alpha$ holding for some formula α.

Defeasible logics such as LDR can also be used to formalise aspects of argumentation [Nut88]. The language of LDR is composed of a set of atoms, and the connectives $\{\neg, \Rightarrow, \rightarrow, >, \wedge\}$. Clauses are formed as follows, where β is a conjunct of literals, and α is a literal: (1) $\beta \rightarrow \alpha$ is an absolute rule; (2) $\beta \Rightarrow \alpha$ is a defeasible rule; and (3) $\beta > \alpha$ is a defeater rule.

An absolute rule cannot be defeated, and the consequent can be inferred when the antecedent is satisfied. For a defeasible rule, the consequent can be inferred when the antecedent is satisfied, and the rule is not defeated. It is defeated when: (1) the complement of the consequent is inferred from an absolute rule; or (2) the complement of the consequent is inferred from a more specific defeasible rule; or (3) there is a more specific defeater rule that has its antecedent satisfied, and has a consequent that is the complement of the defeasible rule. The defeater rules do not give inferences - they can only defeat defeasible inferences.

Example 4.3 *Assume we have a knowledgebase containing just $\beta \Rightarrow \alpha$ and $\beta \wedge \gamma > \neg\alpha$. If we add the literal β, then we infer α. However, if we add the literals β and γ, then we do not infer α, nor do we infer $\neg\alpha$.*

The specificity ordering is determined from the antecedents of the defeasible and defeater rules: If the antecedent to one rule can be used in conjunction with the remainder of the database (i.e. the database minus the two competing rules) to prove the antecedent of the other rule, but not vice versa, then that rule is more specific.

In a similar vein, ordered logic is a defeasible logic but with just defeasible rules and an explicit ordering over formulae [LV90]. It is based on a partially ordered structure of logical theories termed objects. An object is a finite set of rules of the form $\Gamma \rightarrow \alpha$ where Γ is a finite set of literals, and α is a literal. Γ is the antecedent and α is the consequent. A knowledgebase is a tuple $\Delta = (O, \leq, R, K)$ where $(O, \leq)$ is a poset of objects, R is a finite set of rules, and $K : O \mapsto \wp(R)$ is a function assigning a set of rules to each object. The relation $i \leq j$ means that i is more preferred to j.

The rules and facts that are explicitly represented at an object do not constitute the entire knowledge about an object. A specificity relation (i.e. $\leq$) defined on the objects allows for proof rules to infer knowledge at an object on the basis of knowledge higher in the ordering. In particular, the notion of inheritance is supported by allowing knowledge to filter from a higher object to a lower object. Inheritance may be blocked by either being (1) "overruled" by more specific information that is contradictory at the lower object; or (2) "defeat" which occurs when an object i inherits contradictory information from different objects p and q such that there is no preference for one of the objects over the other.

The proof theory for ordered logic is defined in terms of the conclusions that can be proven to hold or proven to not hold at each object. For this, the notation $i.\alpha^+$ means α can be proven to hold at object i, and $i.\alpha^-$ means α can be proven to not hold at object i.

Definition 4.1 *Let $\Delta = (O, \leq, R, K)$ be a knowledgebase, let α be a ground literal, and let $i \in O$ and $\forall j \in O, i \leq j$ holds. The ordered logic consequence relation $\vdash_o$ is defined as follows:*

$$\Delta \vdash_o \alpha \textit{ iff } \Delta \vdash_o i.\alpha^+$$

where

$$\text{[Rule 1]}\quad \Delta \vdash_o i.\alpha^+ \text{ if } \quad \begin{array}{l} \exists p, \text{ such that } i \leq p, \exists \Gamma \rightarrow \alpha \in k(p) \\ \text{such that } \forall \beta \in \Gamma, \Delta \vdash_o i.\beta^+ \\ \text{and } \forall q, \text{ such that } i \leq q \text{ and } p \not< q, \\ \qquad \forall \Phi \rightarrow \neg\alpha \in k(q), \exists \delta \in \Phi, \Delta \vdash_o i.\delta^- \end{array}$$

$$\text{[Rule 2]}\quad \Delta \vdash_o i.\alpha^- \text{ if } \quad \begin{array}{l} \forall p, \text{ such that } i \leq p, \forall \Gamma \rightarrow \alpha \in k(p) \\ [\exists \beta \in \Gamma, \Delta \vdash_o i.\beta^-] \\ \text{or } [\exists q, \text{ such that } i \leq q \text{ and } p \not< q, \\ \qquad \exists \Phi \rightarrow \neg\alpha \in k(q), \forall \delta \in \Phi, \Delta \vdash_o i.\delta^+ \end{array}$$

$$\text{[Rule 3]}\quad \Delta \vdash_o i.\alpha^- \text{ if } \quad \begin{array}{l} \exists \beta_1, .., \beta_n \text{ such that } \beta_1 = \beta_n = \alpha \\ \qquad \text{and } \forall j, \text{ such that } 1 \leq j \leq n \\ \qquad\quad \Delta_K \vdash_o i.\beta_{j+1}^- \text{ implies } \Delta_K \vdash_o i.\beta_j^- \end{array}$$

We can summarise the nature of this definition as follows: (Rule 1) expresses that an inference can be proven at i if there is a rule $\Gamma \rightarrow \alpha$ with all the conditions satisfied, and this rule is not over-ruled or defeated; (Rule 2) expresses that a literal α cannot be proven if for all the rules that imply α, the conditions of the rules are not satisfied or there is an over-ruling or defeating; and (Rule 3) prevents infinite negative looping.

Example 4.4 *Let Δ be the following*

$$\Delta = (O, \leq, \{\{\beta\} \rightarrow \alpha, \{\} \rightarrow \neg\alpha, \{\} \rightarrow \beta\}, \{(o_1, \{\beta\} \rightarrow \alpha), (o_2, \{\} \rightarrow \neg\alpha), (o_3, \{\} \rightarrow \beta)\})$$

where $\{o_1, o_2, o_3\} \subseteq O$, and $o_1 < o_2$, and $o_2 < o_3$ hold. From this we can make the inferences $\Delta \vdash_o \alpha$ and $\Delta \vdash_o \beta$.

Ordered logic has been developed into a general framework, with credulous and skeptical versions [GLV91], and an intuitive fixpoint semantics. Negation-as-failure has been shown to be a special case.

Defeasible logic such as LDR and ordered logic are useful for formalising argumentation. A proof for a conclusion gives the arguments for the conclusion and the defeaters considered give the arguments against the conclusion. Furthermore, these systems incorporate intuitive forms of reasoning with preferences that can be naturally interpreted as preferences over arguments.

Another form of defeasible argumentation in which the acceptability of inferences are considered before they can be used in further derivations is given in [Wag91]. A knowledgebase is a set of classical formulae and a set of rules of the form $\alpha \leftarrow \beta$ where α and β are classical formulae formed from $\wedge$, $\vee$, and $\neg$. Various kinds of reasoning are then proposed including conservative reasoning where

- a conclusion is supported if the knowledgebase contains a rule for it and the premise of which is accepted
- a conclusion is accepted if it is supported and not doubted
- a conclusion is doubted if its contrary is supported

and skeptical reasoning where the following tighter notion of doubt is used

- a conclusion is doubted if its contrary is liberally supported
- a conclusion is liberally supported if the knowledgebase contains a rule for it the premise of which is liberally supported

In order to manipulate the classical formulae, proof rules are defined to allow the derivation of $\alpha \wedge \beta$ from α and β, $\neg\neg\alpha$ from α, $\neg(\alpha \wedge \beta)$ from $\neg\alpha$, and $\neg(\alpha \wedge \beta)$ from $\neg\beta$.

To summarise, defeasible logics capture defeasible rules in the object language that can be put together using the proof theory to construct arguments for and against particular conclusions. Furthermore, these logics offer a range of strategies for resolving conflicts that arise between the competing arguments. This range of strategies offers routes for users to select strategies according to the characteristics of the problem domain. For example, for some applications, perhaps in decision-support, a very skeptical view of inferencing is necessary, whereas in other applications, perhaps in scenario analysis, a more credulous view of inferencing is appropriate. The proposals do vary in notational complexity. Some assume just implication connectives for defeasible rules and some assume the classical connectives in addition. Prototype implementations have been developed for some of these formalisms and it is also possible to use data to learn defeasible rules with priorities [CHS93].

To evaluate defeasible logics with respect to our requirements for reasoning with structured news reports, we see that on the positive side, there is support for ramification analysis, precondition analysis (excluding abduction), and scenario analysis (though undeveloped[1]) in the context of push argumentation. The disadvantage for push argumentation is that some formalisms are limited with respect to capturing preferences over rules. This means that in some situations too many arguments may be generated. Having preferences may allow for some arguments to be eliminated from further consideration (at least explicitly).

Some of the formalisms are also appropriate for pull argumentation. This partly results from the types of inference are in some sense more constrained than in default logics and coherence systems. It seems quite plausible that non-technical users could build knowledgebases of defeasible rules. In some cases, syntactic sugar or graphical notations may be used to make them more amenable to users. We consider this again in Section 5, where we consider a hybrid system, the ArguMed System, developed on top of a defeasible formalism [Ver99]. Also of particular potential is ordered logic which is based on some straightforward principles with notation that could be made much more amenable to a non-technical user group with appropriate syntactic sugar. On the negative side, for defeasible logics there is a lack of richer notation or ontology, and in general a lack of support for abstractions of the knowledge (an interesting exception is the ArguMed System), or facilities for linkage to other sources. Possibly, data mining and machine learning can be used to generalise news reports, and thereby generate knowledge for defeasible reasoning.

4.2 Coalition systems

Coalition systems[2] are based on sets of arguments that defend each other against counter-arguments by banding together for self-defence. A significant proposal that can be described as using coalitions is by Dung [Dun95]. This approach assumes a set of arguments, and a binary "attacks" relation between pairs of arguments. A hierarchy of arguments is then defined in terms of the

[1] We describe scenario analysis as being undeveloped for defeasible logics because the issues raised in Section 3.2 and Section 5.4 have not been formally addressed. Nonetheless, it seems that defeasible logics, coalition systems, coherence systems, and defeasible logics are promising starting points for scenario analysis.

[2] We have invented this appellation to differentiate the argumentation systems discussed in this section from other kinds of argumentation systems discussed in this paper.

relative attacks "for" and "against" each argument in each subset of the arguments. In this way, for example, the plausibility of an argument could be defended by another argument in its subset.

Definition 4.2 *Let S be a set of arguments. An attacks relation is a subset of $S \times S$. A pair $(S, attacks)$ is an* **argumentation framework**. *A set of arguments S is* **conflict-free** *if there are no two arguments A,B in S such that A attacks B.*

Definition 4.3 *An argument A is* **acceptable** *with respect to a set S of arguments iff for each argument B, if B attacks A then B is attacked by some argument in S. A conflict-free set of arguments S is* **admissible** *iff each argument in S is acceptable with respect to S.*

Example 4.5 *Consider a set $S = \{a_1, a_2, a_3\}$, where a_2 attacks a_1 and a_3 attacks a_2. Hence the set $\{a_1, a_3\}$ is admissible.*

The credulous semantics of an argumentation framework is given by the following definition.

Definition 4.4 *A* **preferred extension** *of a pair (S, attacks) is maximal (wrt set inclusion) admissible set of arguments of (S, attacks).*

Definition 4.5 *A conflict-free set of arguments S is called a* **stable extension** *iff S attacks each argument that does not belong to S.*

Example 4.6 *Continuing Example 4.5, the set $\{a_1, a_3\}$ is a stable extension.*

Every stable extension is a preferred extension but not vice versa.

Definition 4.6 *The* **characteristic function** *of a pair $(S, attacks)$, denoted F, is defined as follows: $F(S) = \{A \mid A \text{ is acceptable wrt } S\}$.*

The skeptical semantics of an argumentation framework is given by the following definition.

Definition 4.7 *The* **grounded extension** *of a pair (S, attacks) is the least fixed point of F.*

Finally, the following gives the arguments of an agent who believes everything he can defend.

Definition 4.8 *An admissible set of arguments S is called a* **complete extension** *iff each argument that is acceptable with respect to S, belongs to S.*

A key advantage of Dung's approach is that since it abstracts from the internal form of arguments and from the grounds for attack, they can be instantiated in many different ways. For example, Pollock's approach (discussed in Section 4.1) has been shown to be an instance of preferred semantics [JV00], and Prakken and Sartor have shows that their system (discussed in Section 4.4) is an instance of grounded semantics [PS97].

Dung's framework has also been extended to allow the representation of preferences over arguments. We consider the first such proposal, by Prakken and Sartor [PS97], in Section 4.4. In another proposal, by Amgoud and Cayrol, the definition of attacks is used together with this preference ordering to define an attacks relation, where for arguments a_i, a_j, we have a_i *is defended against* a_j when a_j attacks a_i iff a_i is preferred to a_j. In this sense, preference orderings capture a notion of defence. This defence relation is then used instead of the original attacks relation in revised versions of Definitions 4.2 to 4.8.

Using abduction, Dung's framework has been developed into an abstract argumentation-theoretic framework for default reasoning [BDKT97]. This can be used as a general framework for capturing a variety of approaches including default logic, autoepistemic logic, and logic programming.

Coalition systems are closely related to various developments in logic programming, in particular in stable model semantics. This offers a route to implemented automated reasoning systems for some coalition systems such as [Nie95]. Some coalition systems can also be implemented without recourse to stable model semantics. In [DKT98], a proof procedure for the "admissibility" semantics is given which is more liberal than the stable semantics. This is defined in the form of a logic program.

To evaluate coalition systems with respect to our requirements for reasoning with structured news reports, we see that on the positive side, there is support for ramification analysis, precondition analysis (including abduction), and scenario analysis (though undeveloped) in the context of push and pull argumentation. Given that Dung's framework abstracts away from the internal form of arguments and from the grounds for attacks, it offers an interesting abstraction that is amenable to graphical representation. Though it seems that to more fully meet the need for abstraction, the framework needs to be developed to restrict the arguments shown, and to abstract on the underlying knowledge used to construct the arguments. There is also a need to develop richer notation and ontologies for applications. Possibly, data mining and machine learning can be used to generalise news reports, and thereby generate knowledge for coalition systems.

4.3 Default logics

Default logic was originally proposed by Reiter [Rei80] as a basis of representing and reasoning with default knowledge. It is one of the best known and most widely studied formalisations for default knowledge [Bes89, BDK97, Ant97, Sch98]. Furthermore, it offers a very expressive and lucid language. In default logic, knowledge is represented as a *default theory*, which consists of a set of first-order formulae and a set of *default rules* for representing default information. Default rules are of the following form, where α, β and γ are first-order (classical) formulae,

$$\frac{\alpha : \beta}{\gamma}$$

The inference rules are those of classical logic plus a special mechanism to deal with default rules: Basically, if α is inferred, and $\neg\beta$ cannot be inferred, then infer γ. For this, α is called the pre-condition, β is called the justification, and γ is called the consequent.

The set of formulae that are derivable from a default theory is called an extension. Each extension is a set of classical formulae. There may be more than one extension per default theory.

Default logic extends classical logic. Hence, all classical inferences from the classical information in a default theory are derivable (if there is an extension). The default theory then augments these classical inferences by default inferences derivable using the default rules.

The method for obtaining an extension from a default theory is given in Definition 4.9. This definition introduces the operator Γ that indicates what conclusions are to be associated with a given set E of formulae, where E is some set of classical formulae. In effect, this definition determines whether E is an extension of the default theory.

Definition 4.9 *Let (D, W) be a default theory, where E is a set of classical formulae, D is a set of default rules and W is a set of classical formulae. Let Th be the function that for a set of formulae returns the set of classical consequences of those formulae. For this, $\Gamma(E)$ is the smallest set of classical formulae such that the following three conditions are satisfied.*

1. $W \subseteq \Gamma(E)$
2. $\Gamma(E) = Th(\Gamma(E))$
3. *For each default in D, where α is the pre-condition, β is the justification, and γ is the consequent, the following holds:*

$$\textit{if } \alpha \in \Gamma(E) \textit{ and } \neg\beta \notin E \textit{ then } \gamma \in \Gamma(E)$$

Once $\Gamma(E)$ has been identified, E is an extension of (D, W) iff $E = \Gamma(E)$.

We can view E as the set of formulae for which we are ensuring consistency with the justification of each default rule that we are attempting to apply. We can view $\Gamma(E)$ as the set of conclusions of a default theory: It contains W, it is closed under classical consequence, and for each default that is applicable (i.e. the precondition is in $\Gamma(E)$ and the justification is satisfiable with E), then the consequent is in $\Gamma(E)$. We ask for the smallest $\Gamma(E)$ to ensure that each default rule that is applied is grounded. This means that it is not the case that one or more default rules are self-supporting. For example, a single default rule is self-supporting if the pre-condition is satisfied using the consequent. The test $E = \Gamma(E)$ ensures that the set of formulae for which the justifications are checked for consistency coincides with the set of conclusions of the default theory.

So, if E is an extension, then the first condition ensures that the set of classical formulae W is also in the extension, the second condition ensures the extension is closed under classical consequence, and the third condition ensures that for each default rule, if the pre-condition is in the extension, and the justification is consistent with the extension, then the consequent is in the extension.

Example 4.7 *Let D be the following set of default rules:*

$$\frac{\texttt{company(x)} \wedge \texttt{loss(x,y)} \wedge \texttt{turnover(x,z)} \wedge \texttt{y} < \texttt{z} : \neg\texttt{new-venture(x)}}{\neg\texttt{financially-sound(x)}}$$

$$\frac{\texttt{company(x)} \wedge \texttt{debt(x,y)} \wedge \texttt{capitalization(x,z)} \wedge \texttt{z} < \texttt{y} : \neg\texttt{takeover-target(x)}}{\neg\texttt{good-buy(x)}}$$

$$\frac{\texttt{company(x)} \wedge \texttt{sector(x,y)} \wedge \texttt{growthsector(x,z)} : \texttt{financially-sound(x)}}{\texttt{good-buy(x)}}$$

$$\frac{\texttt{company(x)} \wedge \texttt{sector(x,y)} \wedge \texttt{sector-many-mergers(x,z)} : \texttt{financially-sound(x)}}{\texttt{good-buy(x)}}$$

If W is

$\{$`company(Talk), sector(Talk,telecoms), growthsector(telecoms), new-venture(Talk)`$\}$

there is one extension from (D, W) which is $Th(W \cup \{$`good-buy(Talk)`$\})$.

The generating defaults (the defaults used) for each extension are the basis of the arguments for and against any possible conclusion. In this way, default logic offers an interesting foundation for argumentation. There are also a number of developments that are potentially useful for reasoning with news reports including the Theorist reasoning system, prioritized default logic, and a form of defeasible reasoning in default logic.

The Theorist reasoning system (TRS) [Poo88] is based on Reiter's normal default logic. It has been developed to deal with a range of examples in an intuitive fashion, including manipulation of contradictory multiple extensions. An interesting feature of TRS is the labelling of knowledgebase items to support the inference mechanism.

Labels are associated with formulae according to the following definition, where α and β are classical formulae, γ is a predicate, and γ has the same free variables as β.

$$\mathsf{fact} : \alpha$$
$$\mathsf{default}(\gamma) : \beta$$

Since the argument in a default label is a predicate, then facts can also have this predicate as a sub-formula. This facility is used in TRS to implement a defeating mechanism. A database for TRS is a pair (F, D) where F is a set of facts, and D is a set of defaults.

Example 4.8 *Consider the data $\mathsf{fact} : \delta \rightarrow \neg\alpha$ and $\mathsf{default}(\alpha) : \gamma$. For this data, we obtain γ. However, if we add $\mathsf{fact} : \delta$, then we do not obtain γ because the facts $\{\delta, \delta \rightarrow \neg\alpha\}$ are not consistent with the default formulae β and its label α.*

Definition 4.10 *A scenario of (F, D) is a set $F \cup D'$ where D' is ground and $D' \subseteq D$ and $F \cup D'$ is consistent, where consistency for D' involves both γ and β for all $\mathsf{default}(\gamma) : \beta$ in D'. A TRS extension of (F, D) is a maximal set of logical consequences, under classical inference, of a maximal scenario of (F, D). For TRS, a database (F, D) implies α, denoted $(F, D) \vdash_{TRS} \alpha$, if α is an element in a TRS extension of (F, D).*

The TRS language can be extended to include the following form of labelled formulae, where α is a classical formula.

$$\mathsf{constraint} : \alpha$$

Constraints are used by TRS to reject an extension, but not to augment an extension. For this, a revised definition of a scenario is required.

Definition 4.11 *A scenario of (F, C, D) is a set $F \cup D'$ where D' is ground and $D' \subseteq D$ and $F \cup D' \cup C$ is consistent where consistency for D' involves both γ and β for all $\mathsf{default}(\gamma) : \beta \in D'$. The definition of a TRS extension is amended accordingly.*

Example 4.9 *Take the formulae* $\{\mathsf{constraint} : \alpha \rightarrow \neg\delta, \mathsf{default}(\delta) : \beta \rightarrow \neg\alpha\}$*. If* α *holds, the defeating mechanism prohibits the default labelled with* δ*, but without forcing* $\neg\delta$ *to hold.*

Useful properties of TRS include adding defaults cannot decrease an extension, adding constraints cannot increase an extension, and there is always an extension if F $\cup$ C is consistent.

In prioritized default logic [Bre91], a partial order on the defaults is used. To use this, the notion of a default extension is redefined as follows: If $D_1, .., D_n$ is a collection of sets of normal defaults, where there is a total ordering over these sets, such that D_1 is the most preferred, and W is a set of non-defeasible data, then E is a prioritized default extension of $(D_1, .., D_n, W)$ iff there exist sets of formulae $E_1, .., E_n$ such that

$$\begin{array}{l} E_1 \text{ is an extension of } (D_1, W) \\ E_2 \text{ is an extension of } (D_2, E_1) \\ \vdots \\ E = E_n \text{ is an extension of } (D_n, E_{n-1}) \end{array}$$

In this way the defaults higher in the ordering have preference over the defaults lower in the ordering, when generating extensions. This has been further developed to reason about priorities in the object language [Bre94]. As a result it is possible to change the priorities according to the context. This is a very useful feature for certain domains including for legal reasoning.

There have been a number of other proposals to develop default logic by extending the syntax of default rules. In the approach of Prakken and Sartor [Pra97], a form of defeasible reasoning is based on default rules with an associated label attached. This label is the name for the rule, and it is also a term in the classical language. This means that rules can be referred to in the formulae and hence in the default rules. Arguments are then constructed by composing sequences of default rules and first-order formulae using first-order reasoning. This is building on the notion of default rules as a form of natural deduction rule.

There is a notion of defeat that takes into account the relative priorities of the defaults. The priorities are represented by an ordering over the labels for the defaults. Furthermore, since the labels are terms in the language, the priorities can be reasoned about in the language.

To determine which arguments are regarded as acceptable after taking into account all the possible defeaters, both undercutters and rebutter, the approach incorporates a notion of acceptable arguments that is analogous to that in the coalition systems. In other words, an argument α is acceptable with respect to a set of arguments Φ iff all the arguments defeating α are defeated by an argument in Φ. There is also a version of the Prakken and Sartor framework based on a logic programming language [PS97]. This later version can also be considered as a defeasible logic approach.

There are a number of other approaches based on default logic that could be described as offering argumentative reasoning. The family of default logics is one of the most well-explored formalisms for default knowledge. There is a range of useful variants, and inferencing technology is being developed (for a review see [Sch98]). It is also possible to learn default rules from data [DN99].

There are numerous other non-monotonic reasoning formalisms that may be useful in argumentation. For a review of non-monotonic logics see [Ant97]. We have concentrated on default logic because of its intuitive and practical behaviour with many examples.

To evaluate default logics with respect to our requirements for reasoning with structured news reports, we see that on the positive side, there is support for ramification analysis, precondition analysis (excluding abduction), and scenario analysis (though undeveloped) in the context of push argumentation. However, as they are currently presented the formalisms are really too complicated for pull argumentation, except perhaps Theorist, unless the users have some familiarity with classical logic and default logic. Graphical notations can be used to make them more amenable to users — we return to this in section 5.2 — which can offer abstractions. On the negative side, currently there is a lack of techniques for abstractions, and a lack of support for linkage. There is also a lack of appropriate ontologies for constructing argumentation systems based on default logic.

4.4 Coherence systems

One of the most obvious strategies for handling inconsistency in a database is to reason with consistent subsets of the database. This is closely related to the approach of removing information from the database that is causing an inconsistency. Here, we explore some of the issues relating these approaches in the context of classical proof theory.

In coherence systems, an argument is based on a consistent subset of a inconsistent set of formulae — the inconsistency arises from the conflicting views being represented. Further constraints can be imposed on the consistent subset for it to be an allowed argument. This range of further constraints gives us a range of approaches to argumentation.

Definition 4.12 *Let Δ be a database and let $\vdash$ be the classical consequence relation. Then:*

$$
\begin{aligned}
&\mathsf{CON}(\Delta) = \{\Pi \subseteq \Delta | \Pi \not\vdash \bot\}\\
&\mathsf{INC}(\Delta) = \{\Pi \subseteq \Delta | \Pi \vdash \bot\}\\
&\mathsf{MC}(\Delta) = \{\Pi \in \mathsf{CON}(\Delta) | \forall \Phi \in \mathsf{CON}(\Delta) \Pi \not\subset \Phi\}\\
&\mathsf{MI}(\Delta) = \{\Pi \in \mathsf{INC}(\Delta) | \forall \Phi \in \mathsf{INC}(\Delta) \Phi \not\subset \Pi\}\\
&\mathsf{FREE}(\Delta) = \bigcap \mathsf{MC}(\Delta)
\end{aligned}
$$

Hence $\mathsf{MC}(\Delta)$ is the set of maximally consistent subsets of Δ; $\mathsf{MI}(\Delta)$ is the set of minimally inconsistent subsets of Δ; and $\mathsf{FREE}(\Delta)$ is the set of information that all maximally consistent subsets of Δ have in common. We also have the following relationship.

$$\bigcap \mathsf{MC}(\Delta) = \Delta - \bigcup \mathsf{MI}(\Delta)$$

We can consider a maximally consistent subset of a database as capturing a "plausible" or "coherent" view on the database. For this reason, the set $\mathsf{MC}(\Delta)$ is important in many of the definitions presented in the rest of this section. Furthermore, we consider $\mathsf{FREE}(\Delta)$, which is equal to $\bigcap \mathsf{MC}(\Delta)$, as capturing all the "uncontroversial" information in Δ. In contrast, we consider the set $\bigcup \mathsf{MI}(\Delta)$ as capturing all the "problematical" data in Δ.

Example 4.10 *Let $\Delta = \{\alpha, \neg\alpha, \alpha \to \beta, \neg\alpha \to \beta, \gamma\}$. This gives two maximally consistent subsets, $\Phi_1 = \{\alpha, \alpha \to \beta, \neg\alpha \to \beta, \gamma\}$, and $\Phi_2 = \{\neg\alpha, \alpha \to \beta, \neg\alpha \to \beta, \gamma\}$. From this $\bigcap \mathsf{MC}(\Delta) = \{\alpha \to \beta, \neg\alpha \to \beta, \gamma\}$, and a minimally inconsistent subset $\Psi = \{\alpha, \neg\alpha\}$.*

A problem with using inferences from consistent subsets of an inconsistent database is that they are only weakly justified in general. In other words, if α is an inference from a consistent subset, we do not necessarily have strong reason to believe it. To handle this problem, we can adopt the notion of an argument from a database, and a notion of acceptability of an argument. An argument is a subset of the database, together with an inference from that subset. Using the notion of acceptability, the set of all arguments can be partitioned into sets of (arguments of) different degrees of acceptability. This can then be used to define a class of consequence relations (see for example [BDP93, EGH95]).

Definition 4.13 *Let Δ be a database. An argument from Δ is a pair, (Π, ϕ), such that $\Pi \subseteq \Delta$ and $\Pi \vdash \phi$. An argument is consistent, if Π is consistent. We denote the set of arguments from Δ as $\mathsf{An}(\Delta)$, where $\mathsf{An}(\Delta) = \{(\Pi, \phi) | \Pi \subseteq \Delta \wedge \Pi \vdash \phi\}$. Γ is an argument set of Δ iff $\Gamma \subseteq \mathsf{An}(\Delta)$.*

Definition 4.14 *Let Δ be a database. Let (Π, ϕ) and (Θ, ψ) be any arguments constructed from Δ. If $\vdash \phi \leftrightarrow \neg\psi$, then (Π, ϕ) is a rebutting defeater of (Θ, ψ). If $\gamma \in \Theta$ and $\vdash \phi \leftrightarrow \neg\gamma$, then (Π, ϕ) is an undercutting defeater of (Θ, ψ).*

Rebutting defeat, as defined here, is a symmetrical relation. One way of changing this is by use of priorities, such as in systems based on explicit representation of preference (eg [Bre89, CRS93, BDP95]), or as in systems based on specificity (eg [Poo88]).

For a database Δ, an argumentative structure is any set of subsets of $\mathsf{An}(\Delta)$. The intention behind the definition for an argumentative structure is that different subsets of $\mathsf{An}(\Delta)$ have different degrees of acceptability. Below, we present one particular argumentative structure A^*, and then explain how the definition captures notions of acceptability.

Definition 4.15 *The following sets constitute the argumentative structure A^*, where Δ is a database.*

$$\begin{array}{lcl}
\mathsf{AT}(\Delta) & = & \{(\emptyset, \phi) | \emptyset \vdash \phi\} \\
\mathsf{AF}(\Delta) & = & \{(\Pi, \phi) | \Pi \subseteq \mathsf{FREE}(\Delta) \wedge \Pi \vdash \phi\} \\
\mathsf{AB}(\Delta) & = & \{(\Pi, \phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathsf{MC}(\Delta), \psi \in \Pi \Phi \vdash \psi)\} \\
\mathsf{ARU}(\Delta) & = & \{(\Pi, \phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge \\
 & & \quad (\forall \Phi \in \mathsf{MC}(\Delta) \Phi \not\vdash \neg\phi) \wedge (\forall \Phi \in \mathsf{MC}(\Delta), \psi \in \Pi \Phi \not\vdash \neg\psi)\} \\
\mathsf{AU}(\Delta) & = & \{(\Pi, \phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathsf{MC}(\Delta), \psi \in \Pi \Phi \not\vdash \neg\psi)\} \\
\mathsf{A}\forall(\Delta) & = & \{(\Pi, \phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathsf{MC}(\Delta) \Phi \vdash \phi)\} \\
\mathsf{AR}(\Delta) & = & \{(\Pi, \phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathsf{MC}(\Delta) \Phi \not\vdash \neg\phi)\} \\
\mathsf{A}\exists(\Delta) & = & \{(\Pi, \phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi\}
\end{array}$$

The naming conventions for the argument sets are motivated as follows. T is for the tautological arguments - i.e. those that follow from the empty set of premises. F is for the free arguments - (due to Benferhat et al [BDP93]) - which are the arguments that follow from the data that is free of inconsistencies. B is for the backed arguments - i.e. those for which all the premises follow from all the maximally consistent subsets of the data. RU is for the arguments that are not subject to either rebutting or undercutting. U is for the arguments that are not subject to undercutting. $\forall$ is for the universal arguments - (essentially due to Manor and Rescher [MR70], where it was called inevitable arguments) - which are the arguments that follow from all maximally consistent subsets of the data. R is for the arguments that are not subject to rebutting. $\exists$ is for existential arguments - (essentially due to Manor and Rescher [MR70]) - which are the arguments with consistent premises.

The definitions for $\mathsf{A}\exists$, AF, AT should be clear. We therefore focus on the remainder. AR allows an argument (Π, ϕ) only if there is no maximally consistent subset that gives $\neg\phi$. AU allows an argument (Π, ϕ) only if for all items ψ in Π, there is no maximally consistent subset that gives $\neg\psi$. ARU combines the conditions of the AR and AU. Notice that AR and $\mathsf{A}\forall$ have very similar

$$\begin{array}{c}
\mathsf{AT}(\Delta) \\
| \\
\mathsf{AF}(\Delta) = \mathsf{AB}(\Delta) = \mathsf{ARU}(\Delta) = \mathsf{AU}(\Delta) \\
| \\
\mathsf{A}\forall(\Delta) \\
| \\
\mathsf{AR}(\Delta) \\
| \\
\mathsf{A}\exists(\Delta) \\
| \\
\mathsf{An}(\Delta)
\end{array}$$

Figure 3: Partial order on A^* induced by $\subseteq$. Sets lower in the diagram are subsets of sets higher in the diagram.

definitions, with the only difference being "$\Phi \nvdash \neg\phi$" in AR versus "$\Phi \vdash \phi$" in $\mathsf{A}\forall$. A similar remark applies to AU and AB. Therefore $\mathsf{A}\forall$ and AB are strengthenings of AR and AU, respectively (i.e. "$\nvdash \neg\phi$" replaced with "$\vdash \phi$"). We summarize the relationship between these sets in the diagram in Figure 3. The main features to notice are that A^* is a linear structure, and that there is an equivalence of AF, AB, ARU, and AU.

Example 4.11 *We give an example of a database, and some of the items in each argument set. Take* $\Delta = \{\alpha, \neg\alpha\}$. *Then* $(\{\alpha, \neg\alpha\}, \alpha \wedge \neg\alpha) \in \mathsf{An}(\Delta)$, $(\{\alpha\}, \alpha) \in \mathsf{A}\exists(\Delta)$, $(\{\alpha\}, \alpha \vee \beta) \in \mathsf{AR}(\Delta)$, *if* $\beta \nvdash \alpha$, $(\{\}, \alpha \vee \neg\alpha) \in \mathsf{A}\forall(\Delta)$. *Furthermore,* $\mathsf{A}\forall(\Delta) = \mathsf{AF}(\Delta) = \mathsf{AB}(\Delta) = \mathsf{ARU}(\Delta) = \mathsf{AU}(\Delta) = \mathsf{AT}(\Delta)$.

Example 4.12 *As another example, consider* $\Delta = \{\neg\alpha \wedge \beta, \alpha \wedge \beta\}$. *Then for* $\Pi = \{\alpha \wedge \beta\}$, $(\Pi, \beta) \in \mathsf{A}\exists(\Delta)$, $(\Pi, \beta) \in \mathsf{AR}(\Delta)$, *and* $(\Pi, \beta) \in \mathsf{A}\forall(\Delta)$. *But there is no* $\Pi \subseteq \Delta$ *such that* $(\Pi, \beta) \in \mathsf{AU}(\Delta)$, $(\Pi, \beta) \in \mathsf{ARU}(\Delta)$, $(\Pi, \beta) \in \mathsf{AB}(\Delta)$, *or* $(\Pi, \beta) \in \mathsf{AF}(\Delta)$.

The argumentative logics are, in a key sense, restricted: If a pair of formulae are mutually inconsistent, then none of these argumentative logics will derive any consequences from the conjunction of the two formulae (as an example see Example 4.13). Another significant drawback is the computational complexity of reasoning with argumentative logics since the reasoning is based on consistency checking (though this is also a problem with many of the other argumentation systems).

Example 4.13 *Consider* $\{\alpha \wedge \beta, \neg\alpha \wedge (\beta \rightarrow \gamma\}$. *There is no consistent subset which gives the inference* γ. *In a sense,* γ *is locked into the inconsistency.*

The concept of an argumentative structure, with the two notions of argument and acceptability, are a convenient framework for developing practical reasoning tools.

The types of argument given in Definition 4.15, can be extended with further kinds of argument such as lex arguments [BDP93]. The set of lex arguments is given as

$$\mathsf{AL}(\Delta) = \{(\Pi, \phi) | \Pi \in \mathsf{LEX}(\Delta) \wedge \Pi \vdash \phi\}$$

where

$$\mathsf{LEX}(\Delta) = \{\Gamma \mid \forall\Gamma' \in \mathsf{MC}(\Delta)\ |\Gamma| \geq |\Gamma'|\}$$

The set of lex arguments does not necessarily coincide with that of universal arguments, nor is one a subset of the other in general.

An alternative approach to consistency-based reasoning has led to a more general classical logic framework for argumentation. So far in this review paper, the logic-based formalisms considered incorporate formal representation of individual arguments and techniques for comparing conflicting arguments. In these frameworks, if there are a number of arguments for and against a particular conclusion, an aggregation function determines whether the conclusion holds in the framework. These aggregation functions can be described as being binary since they just consider the existence of arguments for and against, and so they are not sensitive to the number of arguments for or against. In [BH00, BH01], a framework for argumentation with non-binary aggregation functions is presented. In this framework, the language is classical logic, and an argument is a pair where the first item in the pair is a minimal consistent set of formulae that proves the second item. Non-binary aggregation functions can be defined that are sensitive to the number arguments for and against a conclusion. To support this, the framework provides criteria for deciding which arguments to take into account.

Another angle on consistency-based reasoning is the introduction of preference orderings over formulae. In [Bre89], a knowledgebase Δ is a tuple $(\Delta_1, ..., \Delta_n)$ where each Δ_i is a set of classical formulae. Information in Δ_i is preferred to (or more certain than) information in Δ_j if $i < j$. Given a knowledgebase Δ, a preferred subtheory $\Phi \subseteq \Delta$ is obtained using the following definition:

$$\begin{aligned}&\Phi = \Phi_1 \cup ... \cup \Phi_n \text{ is a preferred subtheory of } \Delta\\ &\qquad\text{iff } \forall k (1 \leq k \leq n)\ \Phi_1 \cup ... \cup \Phi_k \text{ is a maximal consistent subset of } \Delta_1 \cup ... \cup \Delta_k\end{aligned}$$

So to obtain a preferred subtheory of Δ, we have to start with any maximal consistent subset of Δ_1, add as many formulae from Δ_2 as consistently can be added and continue the process with Δ_3,...,Δ_n. Reasoning is then done using classical logic with the preferred subtheory.

In [BDP95], another approach to reasoning from an inconsistent knowledgebase is proposed, where each knowledgebase Δ is partitioned into a sequence of disjoint subsets Δ_1,...,Δ_n, and Δ_i is more certain than Δ_j if $i < j$ holds. From these subsets, the following sets can be formed for i $(1 \leq i \leq n)$ $\Delta^i = \Delta_1 \cup ... \cup \Delta_i$. Assuming Δ be a knowledgebase, we can take $\mathsf{FREE}(\Delta)$ as the intersection of all the maximally consistent subsets of Δ. An inference α follows from Δ iff there is a positive integer such that α is a classical inference from $\mathsf{FREE}(\Delta^i)$. This integer provides a qualification on the inferences.

In [CRS93], preferences over individual formulae in a knowledgebase are used to generate preferences over subsets of the knowledgebase. Given a knowledgebase Δ, and a preference relation $<^*$ over Δ, where for $x, y \in \Delta$, $x <^* y$ iff x is less preferable than y. A preference relation over $\wp(\Delta)$ is obtained from $(\Delta, <^*)$ using either the democratism principle, or the elitism principle, defined as follows:

Democratism Let Φ and Ψ be two non-empty subsets of Δ. Φ is democratically preferred to Ψ, denoted $\Psi <^*_{demo} \Phi$ iff for any $\psi \in \Psi \setminus \Phi$, there is a $\phi \in \Phi \setminus \Psi$ such that $\psi <^* \phi$ and $\phi \not<^* \psi$.

Elitism Let Φ and Ψ be two non-empty subsets of Δ. Φ is elitistically preferred to Ψ, denoted $\Psi <^*_{elite} \Phi$ iff for all $\phi \in \Phi \setminus \Psi$, and there is a $\psi \in \Psi \setminus \Phi$, $\psi <^* \phi$ and $\phi \not<^* \psi$.

By analogy with voting theory, democratism prefers that anything removed from a set is replaced by something better, and elitism prefers that anything kept in a set must be better than something removed. Democratism prefers maximal subsets while elitism prefers minimal subsets.

Furthermore, the preferred subtheories of Δ (using the definition of [Bre89]) are democratically preferred subsets of a knowledgebase Δ

Further proposals for resolving conflicts based on obtaining preferences for certain kinds of inconsistent subsets have been made for application in diagnostic systems (see for example [KW87, Rei87]).

A general taxonomy of principles for solving conflicts, such as arising in argumentation, has been proposed [PL92]. This proposal assumes that strategies for resolving conflict can be ordered according to boldness, and this order is independent of the mechanisms for producing conflict or preference. This framework suggests that there are further potentially interesting systems of argumentation to be developed. Whilst this framework is potentially important for all forms of argumentation it seems particularly relevant to consistency-based approaches.

To evaluate coherence systems with respect to our requirements for reasoning with structured news reports, we see that on the positive side, there is support for ramification analysis (excluding abduction), precondition analysis, and scenario analysis (though undeveloped) in the context of push argumentation. However, as they are currently presented the formalisms are really too complicated for pull argumentation, unless the users have some familiarity with classical logic. In some cases, syntactic sugar or graphical notations may be used to make them more amenable to users. On the positive side, data mining and machine learning can be used to generalise news reports, and thereby generate knowledge for coherence systems. On the negative side, there is a lack of richer notation or ontologies for argumentation, and a lack of support for abstraction or linkage.

4.5 Causal networks

Uncertainty and causality are handled in a range of other approaches. In Bayesian networks (for a review see [Jen96, Jen01]), a directed acyclic graph is used to represent causal relations between random variables. These causal relations are used to identify independence assumptions between random variables to facilitate more efficient representation and reasoning with conditional probabilities.

Example 4.14 *Let* `smokes`, `disease`, *and* `symptom` *be random variables and let* $\rightarrow$ *denote a causal influence. Consider the following Bayesian network,*

$$\texttt{smokes} \rightarrow \texttt{disease} \rightarrow \texttt{symptom}$$

From this network, we can obtain the following by assuming `symptom` *is independent of smokes conditional on knowing* `disease` *is true.*

$$P(\texttt{smokes} \wedge \texttt{disease} \wedge \texttt{symptom}) = P(\texttt{symptom} \mid \texttt{disease})P(\texttt{disease} \mid \texttt{smokes})P(\texttt{smokes})$$

Whilst in a sense, Bayesian networks provide a form of argumentation, the nature of the argumentation is fundamentally different from that given by logic-based approaches. In particular, they provide a means for looking at the ramifications of the change in random variables as dictated by the probability distribution and the axioms of probability theory.

For applications where there might not be enough data to form a sufficiently accurate probabilistic model, or for applications where some abstraction of a Bayesian network is desirable, an approach based on qualitative probabilistic networks has been proposed [Wel88, Wel90]. Reasoning in qualitative probabilistic networks (QPN) is dictated by a qualitative probability distribution and the axioms of probability theory, and so it is also fundamentally different from reasoning with logic-based approaches.

$\otimes$	$+$	$-$	0	?
$+$	$+$	$-$	0	?
$-$	$-$	$+$	0	?
0	0	0	0	0
?	?	?	0	?

Table 1: The $\otimes$ operator for combining chains of influence in qualitative probabilistic networks (Taken from [Wel90]).

$\oplus$	$+$	$-$	0	?
$+$	$+$	?	$+$	?
$-$	?	$-$	$-$	?
0	$+$	$-$	0	?
?	?	?	?	?

Table 2: The $\oplus$ operator for combining chains of influence in qualitative probabilistic networks (Taken from [Wel90]).

The QPN approach is based on using the following type of inequality.

$$P(\alpha \mid \beta \wedge \gamma) \geq P(\alpha \mid \neg\beta \wedge \gamma)$$

If this inequality holds, a node in a network corresponding to α is positively influenced by a node corresponding to β. This may be denoted $(i : \beta \rightarrow \alpha : +)$ where i is some label. Similarly, $(i : \beta \rightarrow \alpha : -)$ can be used to represent the following type of inequality.

$$P(\alpha \mid \beta \wedge \gamma) \leq P(\alpha \mid \neg\beta \wedge \gamma)$$

Facts are represented as $(i : \alpha : \lambda)$ where α is a formula formed from disjunction, conjunction, and negation connectives, and $\lambda \in \{+, -, 0, ?\}$. Intuitively, if λ is $+$, then the probability of α is raised, if λ is $-$, then it is lowered, if λ is 0, then it is unchanged, and if λ is ?, then it is unknown.

Such qualitative influences, rather than explicit probability values, can be propagated and aggregated in the network in a way that is consistent with the axioms of probability theory. Whilst the reasoning is weaker than that of normal probabilistic reasoning, it can be useful. Furthermore, it can provide an abstraction of a more complex underlying situation.

Using Table 1, qualitative probabilities can be propagated. Given a fact $(i : \alpha : \lambda_1)$ and a qualitative influence $(i : \alpha : \lambda_2)$, then we obtain the fact $(i : \alpha : \lambda_3)$, where $\lambda_3 = \lambda_1 \otimes \lambda_2$ reading λ_1 from the left column of the table and λ_2 from the top row. An alternative to the $\otimes$ operator is the $\oplus$ operator given in Table 2.

Example 4.15 *Consider the following information.*

f_1 : `lowUnemployment` : $+$
f_2 : `arrogantGovernment` : $+$
f_3 : `lowInflation` : $-$
r_1 : `lowEmployment` $\rightarrow$ `doNotReElectGovernment` : $-$
r_2 : `arrogantGovernment` $\rightarrow$ `doNotReElectGovernment` : $+$
r_3 : `lowInflation` $\rightarrow$ `doNotReElectGovernment` : $-$

From this, we get one argument for $(i :$ `doNotReElectGovernment` $: -)$ *and two arguments for* $(i :$ `doNotReElectGovernment` $: +)$ *where* i *is a label representing the information used in the proof of the argument.*

The qualitative probabilistic reasoner (QPR) is a hybrid approach that integrates logical reasoning offered by QPNs [Par96, Par98, Par01]. QPR provides a formalisation of normative systems for qualitative argumentation. The logical reasoning is a Gentzen-style presentation. As a special case, it offers a probabilistic semantics for the LA logic of argumentation [FKA92].

Some alternative proposals for probabilistic argumentation systems have been based on representing a causal network in propositional logic, and then using a form of Dempster-Shafer theory for probabilistic qualification of the logical inferences [Hae98, HKL00]. This can be viewed as adding a form of probabilistic reasoning to assumption-based reasoning.

Another approach to handling uncertainty and causality is possibilistic networks. In possibilistic networks (for a review see [GK98]), a directed acyclic graph is used to represent casual relations between possibilistic variables. Possibilistic networks provide a form of ramification analysis, though again the nature of the argumentation is fundamentally different from that given by logic-based approaches. In particular, they provide a means for looking at the ramifications of the change in possibilistic variables as dictated by the possibility distribution and the axioms of possibility theory.

To evaluate causal networks with respect to our requirements for reasoning with structured news reports, we see that on the positive side, there is support for ramification analysis, precondition analysis, and scenario analysis in the context of push and pull argumentation. The probability distribution for a causal network allows the probability of ramifications to be determined from preconditions, and for the probability of preconditions to be obtained from ramifications. However, the emphasis is on likelihood (or plausibility for possibilistic systems) of variables rather than on significance of variables. So for example, identification of highly significant, but low probability, ramifications of news reports are not captured by this approach. Furthermore, causal networks offer a graphical representation as an abstraction. In some cases, syntactic sugar may be used to make them more amenable to users.

On the negative side, there is a lack of richer notation in causal networks, and a lack of support for linkage to sources. Furthermore, the type of reasoning is in a sense relatively weak by comparison with the logic-based approaches. In particular, arguments are constructed, but there is no notion of defeat or undercut as in defeasible logics, coalition systems, and coherence systems, and there is no context-sensitive control mechanisms as given in default logics. However, they do capture the likelihood of influence more directly and intuitively than in the logic based approaches. Other advantages of the probabilistic networks approach include the existence of methods and algorithms for generating them from data [Bun96].

5 Hybrid argumentation systems

In analysing formal and informal approaches to argumentation, it is clear there is a range of options for defining functionality. Furthermore, it seems that there are some interesting possibilities for developing argumentation systems that incorporate features from both the formal and informal approaches. This leads to argumentation systems that we may describe as hybrid systems. We discuss some examples of hybrid systems in the following sections.

We believe that in general formal argumentation systems are lacking in a number of areas including (1) lucidity in representation and reasoning including abstractions, graphical representations, and syntactic sugar; (2) argument aggregation; (3) argument ontologies and domain ontologies; and (4) scenario analysis. We discuss these topics in more detail below.

5.1 Lucid representation and reasoning

In order to make logic-based argumentation amenable to end users, we advocate the use of graphical representations of arguments. An example of an informal proposal using graphical representations is cognitive maps [Axe76]. This was proposed as a way to model general situations arising in the social sciences. Another important example is, of course, the approach by Toulmin [Tou58].

In graphical representations, we have a number of choices. The nodes in the graph represent factors that are either decisions that can be made by the user or things in the environment that the user can observe. An arc from a node x to a node y denotes an influence of x on y. In addition, each arc is labelled. The label may be used in various ways including weighting or prioritising arcs relative to the other arcs. Such a graph can be used to model a one-off situation, such as for analysing a strategic decision, or to model a reoccurring situation, such as for analysing a set of operational decisions.

Various logic-based proposals to argumentation can fit into this format. Some, already use a graphical representation (eg qualitative probabilistic/possibilistic networks [Gef96, Par98]). A variety of other approaches can fit indirectly such as some defeasible logics. Different approaches impose different constraints on an argument graph. Some options for interpreting a factor include: it is either "true" or "false"; it is "true", or "false" or "unknown"; it is assigned a possibility value; and it is assigned a qualitative probability value. Similarly there are various options for the arcs, and for argument aggregation functions.

An interesting example of this approach is the ArguMed System[3] developed as an automated argumentation assistance systems for lawyers [Ver99]. The system assists users in making statements, abducing reasons, inferring conclusions, and providing exceptions. There is a graphical representation of attacks of arguments for a propositional logic with syntactic sugar (for example see Figure 4). The underlying formalism is a form of defeasible argumentation which allows for arguments and undercutters to arguments to represented in a form analogous to that originally proposed by Toulmin [Tou58]. The system supports the process of argument construction by allowing for annotation of various kinds of incompleteness including an exception that is not justified and undercutter that is not justified. The ArguMed software incorporates algorithms for checking argument structures are complete. It also incorporates various editing functions. Overall the system currently offers one of the best examples of a hybrid system.

Another interesting example is the StAR Demonstrator developed as a prototype computer-based assistant for the prediction of potential carcinogenic risk due to novel chemical compounds [KAEF95a]. This system constructs arguments for and against ascribing various carcinogenic risk classifications on compounds. The system draws on heterogeneous data sources and gives the results using graphical and textual presentations. The argumentation is based on a variant of argumentative logics given in Section 4.4 [EGKF93]. In addition, quantitative reasoning in the form of Dempster-Shafer theory is incorporated into the argumentation [KAEF95b]. Some of these ideas have fed into the PROforma[4] method and tools which also incorporates a form of argumentation [FD00].

An application of the PROforma methods and tools has been in decision support for clinicians selecting drugs for prescription. An argumentation approach is taken which looks at the patient record for evidence of problems with particular drugs and other drugs being taken, and at a drugs directory for information on appropriate drugs for a particular disorder and for information on possible side effects and contra-indications. Arguments are then constructed for and against each possible drug. All the information is presented to the doctor, and so argumentation is a way of presenting the information, and a way of highlighting information that may inadvertently be

[3] ArguMed System: www.metajur.unimaas.nl/~bart/aas

[4] PROforma: www.acl.icnet.uk

```
Peter has committed a tort.
  ^
  |
  |  THAT There is a ground of justification for Peter's act,
  |  IS AN EXCEPTION TO THE RULE THAT,
  |  IF Peter has violated a property right,
  |  THEN Peter has committed a tort.
  |    ^
  |    |
  |    |  According to art.6:162.2 of the Dutch civil code,
  |    |  violations of property rights are torts,
  |    +--unless there is a ground justification.
  |
  x--There is a ground of justification for Peter's act.
  |
  |
  +--Peter has violated a property right.
```

Figure 4: An argument structure represented in the ArguMed System [Ver99]. The conclusion `Peter has committed a tort` that is derived from the fact `Peter has violated a property right` has the warrant `IF Peter has violated a property right, THEN Peter has committed a tort` with backing `According to art.6:162.2 of the Dutch civil code, violations of property rights are torts, unless there is a ground justification`. In addition, there is the undercutter `There is a good of justification for Peter's act` obtained by the exception to the warrant.

overlooked [FJR98].

Another example of a system that has been motivated by the need for pull argument is the Zeno Argumentation Framework [GK97]. This system is designed to help users analyse questions, so called "issues", by identifying choices and constraints for solving each issue, and by identifying pros and cons for justifying the choices and constraints considered. The result of using the system is that the user has constructed a tree with the main issue at the root, and then each subsequent node is a choice, a constraint, a subsidiary issue, a pro or a con. The leaves are assumed to be undisputed facts. Each such fact is a pro, a con, or a constraint. An example of a graph produced in this way is given in Figure 5. Different formalised criteria can be used for drawing a conclusion from a Zeno argumentation graph including: Beyond reasonable doubt; Best choice; No better alternative; Preponderance of evidence; and Scintilla of evidence.

Further examples of pull argumentation systems with a lucid representation include the Room 5 system [LNA+97]. This was implemented as a system for pull argumentation with US supreme court precedents. It is partly based on formal argument games and defeasible logics, and it also supports linkage of arguments to sources.

There are various other graphical tools either distributed by academic groups, or companies, such as Belvedere[5], QuestMap[6], and Assistum[7], that incorporate the facility to represent and edit pros and cons for arguments and various annotations such as representing assumptions, causal explanations, predictions, and questions. However, these tools do not appear to incorporate a sufficiently sophisticated or formal framework for logic-based inferencing for domains such as handling news reports.

[5] Belvedere: www.advlearn.lrde.pitt.edu/belvedere
[6] Questmap: www.softbicycle.com
[7] Assistum: www.assistum.com

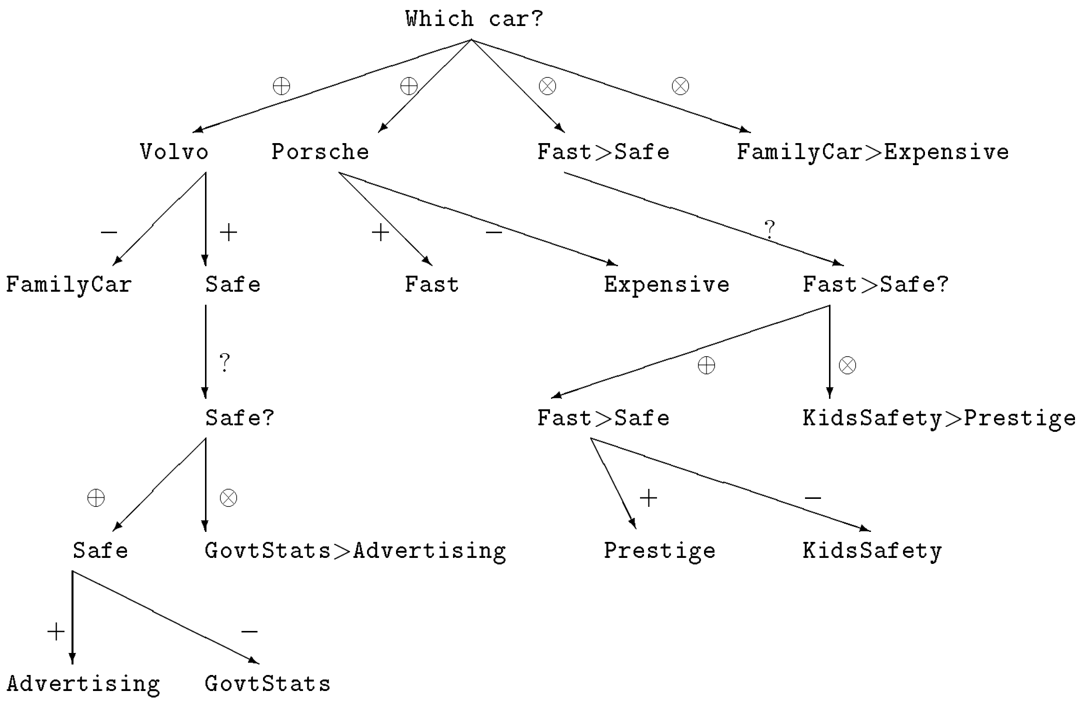

Figure 5: An example of a Zeno argumentation graph where ? denotes an issue, $+$ denotes a pro, $-$ denotes a con, $\oplus$ denotes a choice, and $\otimes$ denotes a constraint. In this example, the main issue is `which car?`. There are two choices `Volvo` and `Porsche` and two constraints: `Fast` is preferred to `Safe`; and `FamilyCar` is preferred to `Expensive`. There are pros and cons for the choices. Of these pros and cons, the only issue is `Safe?`. The others are accepted as undisputed facts. The only choice for the issue `Safe?` is `Safe` and there is a pro and a con, both of which are accepted as undisputed facts. However, there is also a contraint that `GovtStats` (Government Statistics) are preferred to `Advertising`. In an analogous way, the constraint `Fast` is preferred to `Safe` is also an issue. This constraint can be rejected because the con `KidsSafety` is preferred to the pro `Prestige`. This example is adapted from [GK97].

From the examples of formal approaches to argumentation, such as PROfroma, the Zeno system, and the ArguMed system considered in this section, the causal networks considered in Section 4.5, and the causal mapping and WeFA graph techniques considered in Section 5.2, graphical representations can greatly increase the lucidity of the representation and reasoning involved in argumentation. It would seem that there are further possibilities for graphical representations drawing further features of the underlying argumentation. Furthermore, from these proposals it seems that many of the logic-based approaches considered in Section 4 can be integrated with graphical abstractions. In a similar vein, it would seem that syntactic sugar, as illustrated by the ArguMed system could be developed for other formal logic-based approaches. However, with both the graphical representations and syntactic sugar, there is the question of ontology that needs further consideration. We look at this in Section 5.3.

5.2 Argument aggregation

Most proposals of logic-based argumentation adopt a single definition for aggregating a number of arguments: For example, α holds if α is not rebutted and there is an argument for α that is not undercut. Some exceptions to this are: (1) Non-binary argument aggregation functions proposed in [BH00, BH01]; (2) A form of defeasible logic that incorporates notions of argument accrual, where arguments for a conclusion can reinforce each other, and compound defeat, where groups of arguments can defeat other groups of arguments [Ver95]; and (3) Reason-based logic that extends classical logic with principles for manipulating sets of reasons for and against conclusions such as for example "if a set of reasons for a conclusion outweigh a set of reasons against a conclusion, then a superset of the first set will outweigh the second set" [Hag97, VHvdH98].

As another alternative, we may wish to associate the aggregation functions with the knowledge and so use a variety of aggregation functions. In other words, we may want the choice of argument aggregation function to be context dependent. So in a graphical representation, rather than adopt one aggregation function for a graph, we want to allow a user to specify aggregation functions for each node in an aggregation graph. And in a logical representation, the aggregation function is implicit in the object-level rules in the knowledgebase.

To illustrate this, we can consider a formalisation of ramification analysis. This is based on a notion of causal maps for modelling "causal relationships" between events. Events are either existential event classes, for example `presidential-election`, with instances being *true, false*, or *unknown*, or directional events classes, for example `inflation`, with instances being *increasing, decreasing* or *unchanging*. Each causal map is a directed acyclic graph, where each node represents an event, and each arc represents a causal relationship.

Using causal maps, new information can be propagated to determine possible ramifications. These ramifications are also described in terms of events. Ramifications are represented by arcs between events. So for example in Figure 6, if we have the news that the `number of aircraft sold` is true, then we can infer that the `unit production cost of aircraft` has decreased.

As an example of the need for context dependent argument aggregation consider Figure 6. We may wish to have an aggregation function for the node `unit production cost of aircraft` that takes the value of "decrease by 20%" if the `number of aircraft sold` takes the value "increases by 30%" and `cost of aircraft development` takes a value "increases by less than 10%". Clearly, we need to extend this definition to take account of all possible ranges of values that could be updated by the news. But at least, it illustrates how we can specify much more detailed values for the nodes, so moving from just "true" and "false" assignments, and for propagating these values.

Whilst causal maps offer a lucid view on ramifications, there is a need to support automated reasoning, to address problems of incompleteness, to represent further conditions on ramifications,

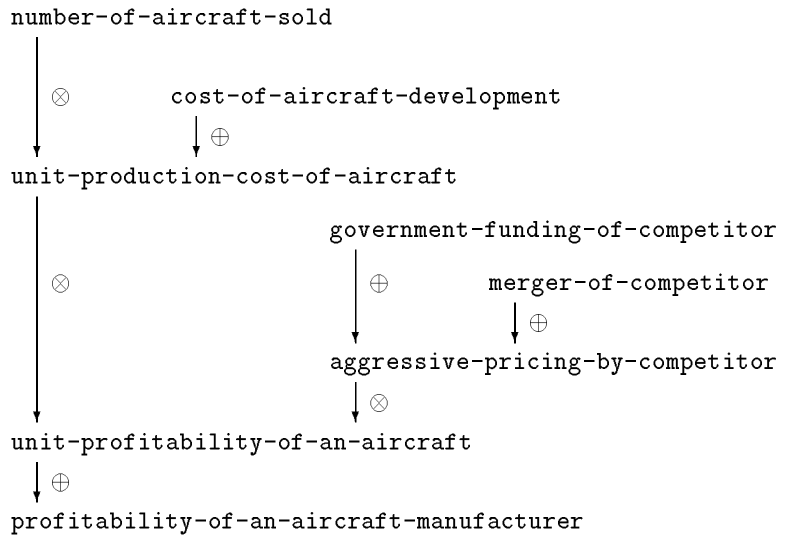

Figure 6: A causal map showing the likely ramifications of news on the profitability of an aircraft manufacturer. Each node denotes a factor that increases in value (or becomes true), or decreases in value (or becomes false), or remains constant. A scenario is an assignment of "increases", "decreases", or "constant", to each node. An $\oplus$ arc is a monotonically increasing influence and an $\otimes$ arc is a monotonically decreasing influence. So for an $\oplus$ arc from X to Y with no further arcs entering Y, if X is assigned "increases", then Y is inferred as "increases". However, for nodes with more than one incoming arc, resolution strategies are required for determining the updated values [Hun00b].

and in particular to address the problem of argument aggregation. To do this causal maps are translated into default logic. In this way, a causal map is an abstraction of a default theory.

To illustrate, one of the ways that a part of the causal map can be translated into a default theory, consider the potential conflict arising if both `number of aircraft sold` and `cost of aircraft development` are both raised in value. The facts can then be represented by the following.

`number-of-aircraft-sold(INCREASED)`   `cost-of-aircraft-development(INCREASED)`

In the default rule (below), if it is possible to infer `inflation-in-development-costs(X)` and $\mathtt{X} < \mathtt{10\%}$ for `X`, then the conflict is resolved. Obviously further rules are needed for when the justification is not satisfied, and further rules are required for the rest of the graph. An outline methodology is presented in [Hun00b] for developing these default theories.

$$\frac{\begin{array}{l}\mathtt{number\text{-}of\text{-}aircraft\text{-}sold(INCREASED)}\\ \wedge\mathtt{cost\text{-}of\text{-}aircraft\text{-}development(INCREASED)}\\ \wedge\mathtt{inflation\text{-}in\text{-}development\text{-}costs(X)} \wedge (\mathtt{X} < \mathtt{10\%})\\ :\mathtt{unit\text{-}production\text{-}cost\text{-}of\text{-}aircraft(INCREASED)}\end{array}}{\mathtt{unit\text{-}production\text{-}cost\text{-}of\text{-}aircraft(INCREASED)}}$$

Another approach to argument aggregation is found in Weighted Factors Analysis (WeFA), a form of pull argumentation. This is an approach for elicitation, representation, and manipulation of knowledge about a given problem being analysed [Hes99]. For example the problem might be to ensure that `The project is completed on time and within budget` or that `The business has increased profitability`. Central to this proposal is that a group of experts in the area of the problem can identify factors with positive or negative influence on the problem outcome. Furthermore, for each of these factors, this breakdown can be repeated by recursion.

The tangible output of WeFA is a WeFA graph. This is a set of nodes denoting factors that can directly or indirectly influence a factor of interest in the graph. Each directed arc is a direct influence of one factor on another. A chain of directed arcs indicates an indirect influence. The influences may be identified as either positive or negative. For example, `sales` and `costs` are two factors that influence the factor of `profitability` in an organisation, where `sales` has a positive influence on `profitability` and `costs` has a negative influence on `profitability`.

In addition to identifying the polarity for each influence, it is possible to identify the relative weight of each influence. For a factor in a graph, there may be a number of factors that influence it. These factors may have different weights. For example, consider the factor `passing a new bill in parliament`, with the subsidiary factors `the bill is supported by a backbench MP` and `the bill is supported by the Prime Minister`. Both these subsidiary factors have a positive influence, but the latter factor carries much more weight than the former.

Binary WeFA which is a variant of WeFA where the factors in the graph are restricted to being either true or false [HH01]. Imposing this restriction on a WeFA graph allows for the approach to be formalised by translating the graph-based reasoning into a form of defeasible logic. This incorporates an adaptation of rule engineering techniques from [HM98].

In order to do reasoning with a Binary WeFA graph, a scenario is assumed which is an assignment of true or false to some of the nodes in the graph. If a node is true, then any arc coming out from that node is fired. From this assignment, we can propagate truth assignments to the other nodes as follows: For each node x, if the sum of the weights on the positive fired incoming arcs is greater than the sum of the weights on the negative fired incoming arcs, then node x is true, otherwise it is false. The gives a way of representing and reasoning with the relative significance of direct influences on a node.

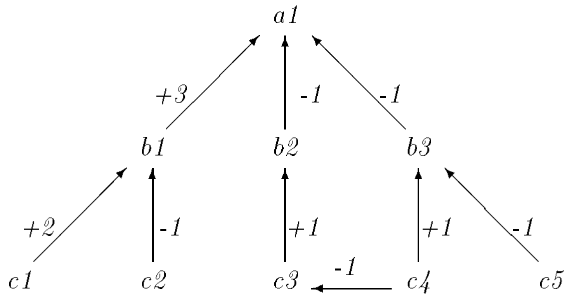

Figure 7: A Binary WeFA graph with the factor of interest being $a1$. $b1$ has a direct positive influence on $a1$, and $b2$ and $b3$ have direct negative influences on the aim. $c1$, $c4$ and $c5$ have an indirect positive influence on $a1$, and $c2$, $c3$ and $c4$ have an indirect negative influence on $a1$. As an example of a scenario, we could have $c1$, $c2$ and $c3$ true. From this, we would get $c4$ and $c5$ false by default, and $b1$, $b2$, and $b3$ true by inference, and finally $a1$ true by inference.
.

Both causal mapping and binary WeFA graphs offer a way for users to become involved in the development of aggregation functions. These ideas could be extended to a wider range of graphical notations and underlying formalisms.

5.3 Richer ontologies for argumentation

Here, we use the notion of an ontology to mean the set of constructs used to capture some knowledge in a knowledgebase. In other words, we may regard an ontology as delineating the basic components of a language for an application. The importance comes when we want languages that more closely meet users needs. If we have appropriate ontologies then users can represent and reason with information more easily.

Currently in argumentation there is a gap between what the formalisms offer and what users need. For this, there are two types of ontology that need further consideration. These are argument ontologies and domain ontologies. We define these below, and then consider examples.

Argument ontologies These can be regarded as meta-level ontologies. Essentially, there is a need for more meaningful constructs for the components of argumentation. Components such as reasons, defeaters, undercutters, etc may be too abstract for many users. So in a given application area different types of defeater, warrant, etc, can be recognised and possibly used differently.

Domain ontologies These can be regarded as object-level ontologies. Essentially, there is a need for domain axioms to be defined that can be incorporated in knowledgebases of different argumentation systems. This would enable argumentation systems to be developed more quickly and for richer domain specific properties to be identified. Therefore it would seem that the motivation for domain ontologies in argumentation are the same as those for using domain ontologies in artificial intelligence in general.

The ArguMed System [Ver99] is a nice example of an argument ontology. Perhaps law lends itself to a richer, more meaningful, argument ontology. The basic components of arguments (undercutters, rebutters, exceptions, etc) are labelled with syntactic sugar to indicate in the role taken in the legal argument. This ontology is effectively defined in terms of the underlying defeasible logic. Further examples of argumentation systems for legal reasoning are covered in [Hag97, Pra97].

A second example is an ontology for rhetorical arguments [Sil96, Sil97]. Rhetorical argumentation incorporates a wider range of influences than normally seen in logical argumentation. For example, the basis for arguments include reciprocity (return a favour or a hurt), fairness (equal treatment for equal cases), and benefit (do x because x has good benefits). These types of argument and influence are then used in addition to logical arguments in an extended version of Toulmin's framework [Tou58]. A significant advantage of this proposal is that a much wider variety of arguments can be compiled and their relationship delineated clearly in a graphical form. This is appealing for pull argumentation systems and for abstractions of knowledgebases in push argumentation systems.

The problem with this approach is incorporating a formal definition for reasoning in this system. Representing influences can be useful, but really we want to undertake inferencing. This requires definitions for aggregation and propagation of arguments. For example, consider the following arguments which are of different rhetorical types.

`X should not do Y because Y is illegal` (Legality argument)

`X should do Y because Z asked X to do Y and X owes Y a favour` (Debt argument)

In these rhetorical arguments, it is unclear which should take precedence, or what other information we are allowed to assume to resolve the conflict. Whilst it is useful to flag the different rhetorical characteristics of these arguments, we really need to formalise these so that we can formalise aggregation and hence inferencing. Indeed, in some respects the framework of Prakken and Sartor [PS97, Pra97] which was discussed in Section 4.3 offers a route for this. But maybe we need more. Maybe we should seek a rich axiomatization for defining the relationships that can hold between the constructs proposed in rhetorical argumentation.

A third example is the argument ontology for potential hazards, such as health risks from genetically modified food, and the effects of smoking [MP99]. The types of inferences seem much more meaningful than normally used in a logical reasoning system, and are described in terms of the types of evidence used. Examples include:

```
FROM A limited number of cases TO Cumulative exposure

FROM A pure chemical substance TO A Chemical compound

FROM Localized exposure TO Broader exposure
```

The idea is that for a set of twenty of these inference rules, arguments can be constructed for and against some chemical or biological substance or agent being a hazard. In a sense, each of these rules can be viewed as a meta-level rule in a way that is analogous to default rules in default logic.

Again there is a need for axioms for relating these different types of inference or perhaps they could be used for some kind of type-checking for arguments. Having a richer ontology creates problems when it comes to automated reasoning. For example how do you have a general-purpose aggregation function for combining a collection of the inferences that are listed in the hazard ontology? This is not necessarily a barrier to developing the ideas since there is value in just representing arguments and not actually having automated reasoning. But the ontologies do need to be formally-based, and a semantics is desirable.

A fourth example is from the informal logics community, and more generally in philosophy, where there is a substantial literature on the nature of argumentation that looks at questions like "what is a strong argument", and "what is a valid argument". This literature offers numerous directions relevant to developing argument ontologies, though we will just consider "appeals to authority" (*ad verecundiam*) here. In this, we equate "authority" with "expert authority", rather than "administrative authority". The former is authority based on knowledge, whereas the later is authority based on power.

Appealing to authority can be justifiable. For example, having a medical operation done is normally justifiable when recommended by a clinician, but it is not normally justifiable when recommended by say an architect. However, a difficulty with appealing to authority is determining when it is justifiable. In [Wal89], various suggestions are made for when appealing to authority is appropriate, and which we attempt to synthesise into the following criteria.

- The expert is suitably qualified in the field.
- Ignoring the expert's advice may not be feasible or wise.
- The expert offers sufficient confidence in the advice.
- None of the expert's caveats have been overlooked.
- The expert's advice seems the best out of any competing sets of expert's advice.

These criteria suggest a meta-level formalisation of reasoning about experts and their advice could be feasible, and that inferences from such a formalisation could affect the object-level reasoning with arguments. Already some of these features are available in a form of modal logic [Cho95].

From these examples, it seems argument ontologies are an interesting and viable development. Perhaps the most important goal in the short term is the development of argument ontologies that are graphically-based and the ontologies are formalised in an underlying logic. This would allow richer and more meaningful abstractions to be obtained.

The development of domain ontologies is a central issue in artificial intelligence, and much of the research in knowledge representation and reasoning is in some respects closely interlinked with that in ontologies. Whilst much of the focus on ontologies in knowledge representation and reasoning has been on description logics, it is clear that a much wider range of logics are relevant to the aim of building domain ontologies.

As an example of the need to develop domain ontologies in argumentation, consideration needs to be given to arguments over time. Whilst sets of arguments evolve over time, influences become weaker or stronger, warrants appear and disappear, and so on, argumentation systems tend to be atemporal. Yet within AI and logic there is an extensive literature on temporal reasoning that could be harnessed for temporal argumentation systems to model arguments over time (for a review of temporal logics see [GHR94]). Already time is a parameter that has been used in a dialectical framework [MP01]. Though the notion of time in this framework is limited to providing a linear sequence over the receipt of items of information that can be used in argumentation. Potentially, harnessing temporal logic offers much more than this in modelling the argumentation process.

Another issue is that important data and knowledge is temporal in nature, such as **`always submit your tax form before the end of the next tax year or you will be fined`**. There is much potential in using a temporal language as the object language. So if we are considering the pros and cons for some formulae α, then this formula may be a temporal logic formula, and the assumptions used may also be temporal formulae [Hun01c].

Even within the structured news reports, there are temporal issues such as the granularity and format for the temporal information to handle. To illustrate some of our needs, consider a set of news reports in the following situations.

- The set of reports refer to the same time point of a subject. As an example, consider a report from a newspaper and a report from a radio station on the weather in London today. So the weather in London is the subject, and the time point is today. However, the TV report and radio reports have not necessarily been made at the same time. So for example, the newspaper report could have been published three days ago, whereas the radio report could have been broadcast today.
- A set of reports refer to the same time intervals of a subject. As an example, consider a report from a TV station on the weather in London today and a report from a radio station on the weather in London today. So the weather in London is the subject, and the subject time interval is today. As with the example above, the TV report and radio reports have not necessarily been broadcast at the same time.
- A set of reports are broadcast at the same time point. As an example, consider a radio broadcast today with a long-range weather report for weather in Europe over the next month, and a TV broadcast today with a weather report for London today. Here, the granularity of the periods of the subjects of the reports is different.
- A set of reports are broadcast during the same time interval. As an example, consider a radio broadcast today with a weather report for weather in Europe over the next month, and a TV broadcast yesterday with a long-range weather report for weather in Europe over the next week. Here, the granularity of the periods of the subjects of the reports is different.

To handle these, and a number of related issues, we need to consider how we can enhance our approach to reasoning with the temporal aspects of news reports [Hun01b]. These issues obviously impinge on the ontology for time in argumentation, and raise important questions of how richer temporal reasoning can be supported.

Other aspects of developing domain ontologies could draw on options of using ontologies for XML documents (see for example [ES01]), using comprehensive semantic networks such as WordNet [Mil95], using domain specific semantic networks (see for example [CTHD00]), and using machine readable dictionaries for representing and reasoning with lexical knowledge (for a review see [WSG96]).

Developing richer ontologies for formal approaches to argumentation seems like a potentially important step in increasing the uptake of formal approaches in applications. There are a number of papers that touch on the topic of ontologies in the informal argumentation camp and it may be fruitful to find more suggestions for enhancing formal proposals.

5.4 Scenario analysis

In an argumentation system, there may be a number of factors represented that can directly or indirectly influence the factors of interest. So the outcome of the factors of interest can be regarded as contingent on the value of the other factors in the knowledgebase. To analyse this contingency, there is a need for scenario analysis. Different scenarios tell us different things about the problem being analysed. Of particular interest are scenarios that could be described as extreme or normal such as:

- Best-case scenarios

- Worst-case scenarios
- Average-case scenarios
- Prototypical-case scenarios

An average-case scenario is a scenario that is one of the most likely scenarios whereas the extreme case scenarios are not necessarily likely. Given that it would normally take too long to exhaustively consider all scenarios using key scenarios such as these is important. There is also the option of working backwards in argumentation: Starting with a particular outcome for a factor of interest, the aim is to identify which scenario leads to that outcome.

In addition to considering individual scenarios, there is the need to consider sets of scenarios. These can reveal the sensitivity of the knowledgebase or inferencing to changes in the scenario. If the behaviour of the outcome appears very sensitive to the input, then this may indicate that the deciding factors for the desired outcome of factors of interest have not been adequately established. This kind of scenario analysis is called sensitivity analysis. A specialisation of sensitivity analysis is to order the scenarios according to likelihood. If the truth (or falsity) of a node, is relatively stable in inferred truth value, then there is increased confidence that uncertainty has been minimised in the model.

Another kind of scenario analysis is temporal analysis. Here a sequence of scenarios is meant to indicate how inputs to a problem may evolve over time. For example if there is a knowledgebase that models a project management problem, there may be a sequence of scenarios S1, S2, S3, ... where S1 is the scenario for month one, S2, is the scenario for month two, and so on. Here, the aim is to see how the outcome is affected by the sequence of scenarios.

In a logic-based approach a scenario can be represented by a set of formulae, perhaps just a set of ground literals, which are added to the knowledgebase. However, the suggestions in this section indicate that there are a number of interesting directions in which logic-based scenario analysis could be developed. A key challenge is the identification of extreme and normal scenarios.

The informal description of scenario given in this section does not assume a total order over scenarios. It can be a partial order and it does not assume there is a unique best case or unique worst case. If it can be a partial order, then it may support multi-dimensional means of comparing scenarios. Of course, this then means the research may overlap with a number of other areas such as voting theory and utility theory.

6 Conclusions

We believe that there is a great need for tools for argumentation to support users in dealing with conflicting information. In a number of areas, such as the medical and legal domains, there has been some convincing progress. Here we have considered analysing structured news reports as another application area where argumentation tools could be particularly useful.

In order to develop tools, we have considered a variety of proposals, both formal and informal, for argumentation systems. From this, we have noticed that there are desirable features in both the formal and informal camps. In this paper, we have attempted to flag some for these desirable features, and then suggest that we may need hybrid argumentation systems that combine a number of these features within a particular argumentation system.

Returning to the requirements that we considered in Section 3.2, for developing argumentation systems for structured news reports, we can make the following observations.

Ramification analysis The proposal for ramification analysis based on causal mapping and default logic [Hun00b], raises the possibility of other underlying logical formalisms being integrated with casual mapping, or a variant of this graphical language, for conducting ramification analysis. Of particular interest would be an approach based on non-binary aggregation functions [BH00, BH01]. Many of the formal proposals considered in Section 4 are also potentially applicable in this task.

Precondition analysis Binary WeFA offers a pull argumentation proposal [HH01] that could be developed for precondition analysis with structured news reports. However, for push argumentation, formal proposals derived from abductive inference frameworks, such as [KKT93], may be more appropriate. Coalition systems may be a good starting point given the established relationships with abduction [BDKT97, DKT98].

Scenario analysis From the discussion in Section 5.4, this is an underdeveloped area of logic-based argumentation. Yet, it seems an interesting and viable goal to develop logic-based scenario analysis for structured news reports.

Abstraction of knowledge There are graphical techniques that can be harnessed for abstracting from logic-based reasoning, as illustrated by the discussions in Section 5.1. Furthermore, the possible graphical notations available for analysing news reports could be extended as discussed in Section 5.3 using appropriate argument ontologies. There is also a need to develop other techniques that allow presenting only key arguments to the user and techniques that allow a simplified version of the knowledge to be presented to the user.

Linkage to sources Connections between graphical abstractions of argumentation and textual explanations are appealing for systems for analysing news reports. In the short term, this seems more of an implementation issue rather than a question of adapting the underlying formalism. In the longer term, it should be possible to interact more directly with the argumentation process. For example, this may include linking two or more knowledgebases that offer different perspectives, or abstractions, on the same topic, and being able to run these in parallel to compare and contrast the resulting reasoning.

Synthesis of news Much textual information in the form of reports, papers, editorials, etc, is available that could be described as being a synthesis of news. The development of tools for decomposing such sytheses may prove useful in knowledge engineering, and for abstraction and explanations of knowledge, used in argumentation. A starting point may be rhetorical structure theory [MT88].

Generalisation of news There are numerous well-established data mining/machine learning techniques that could be deployed to identify useful patterns in sets of structured news reports. The main question is how viable is it to base argumentation systems on knowledge obtained in this way?

We would like to stress that we believe that there is no one argumentation system that will meet all needs. Rather, there are tools for jobs that should be selected according to the needs of the problem. Furthermore, we believe that it is imperative to adopt formal foundations for any argumentation system. Otherwise, there is ambiguity and unnecessary confusion over the notation, and potential problems in the reasoning, and so rendering it infeasible to engineer a system with sufficient knowledge to be useful, and/or leading to a lack of trust in the system by the user.

7 Acknowledgements

The author would like to thank Peter McBurney for helpful feedback on an earlier version of this paper, and the reviewers for numerous comments and suggestions for improvement.

References

[Abi97] S Abiteboul. Querying semi-structured data. In *International Conference on Database Theory*, pages 1–18, 1997.

[Ant97] G Antoniou. *Non-monotonic Reasoning*. MIT Press, 1997.

[Axe76] R Axelrod, editor. *Structure of Decisions: The Cognitive Maps of the Political Elites*. Princeton University Press, 1976.

[BDK97] G Brewka, J Dix, and K Konolige. *Non-monotonic Reasoning: An Overview*. CLSI Publications, 1997.

[BDKT97] A Bondarenko, P Dung, R Kowalski, and F Toni. An abstract argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[BDP93] S Benferhat, D Dubois, and H Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 1449–1445. Morgan Kaufmann, 1993.

[BDP95] S Benferhat, D Dubois, and H Prade. A logical approach to reasoning under inconsistency in stratified knowledge bases. In Ch Froidevaux and J Kohlas, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'95)*, volume 956 of *Lecture Notes in Computer Science*, pages 36–43. Springer, 1995.

[Bel77] N Belnap. A useful four valued logic. In G Epstein, editor, *Modern Uses of Multiple-valued Logic*. Reidel, 1977.

[Bes89] Ph Besnard. *An Introduction to Default Logic*. Springer, 1989.

[BH00] Ph Besnard and A Hunter. Towards a logic-based theory of argumentation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI'2000)*, pages 411–416. MIT Press, 2000.

[BH01] Ph Besnard and A Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.

[Bre89] G Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *Proceedings of the Eleventh International Conference on Artificial Intelligence (IJCAI'89)*, pages 1043–1048, 1989.

[Bre91] G Brewka. *Non-monotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press, 1991.

[Bre94] G Brewka. Reasoning about priorities in default logic. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94)*, pages 420–425. MIT Press, 1994.

[Bun96] W Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8:195–210, 1996.

[Bun97] P Buneman. Semistructured data. In *Proceedings of the ACM Symposium on Principles of Database Systems*, 1997.

[Cho95] L. Cholvy. Automated reasoning with merged contradictory information whose reliability depends on topics. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU)*, Fribourg, July 1995.

[CHS93] J Cussens, A Hunter, and A Srinivasan. Generating explicit orderings for non-monotonic logics. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93)*, pages 420–425. MIT Press, 1993.

[CL96] J Cowie and W Lehnert. Information extraction. *Communications of the ACM*, 39:81–91, 1996.

[CML01] C Chesnevar, A Maguitman, and R Loui. Logical models of argument. *ACM Computing Surveys*, 32:337–383, 2001.

[CRL00] D Carbogim, D Robertson, and J Lee. Argument-based applications to knowledge engineering. *Knowledge Engineering Review*, 15:119–149, 2000.

[CRS93] C Cayrol, V Royer, and C Saurel. Management of preferences in assumption based reasoning. In *Information Processing and the Management of Uncertainty in Knowledge based Systems (IPMU'92)*, volume 682 of *Lecture Notes in Computer Science*. Springer, 1993.

[CTHD00] P Clark, J Thompson, M Holmback, and L Duncan. Exploiting a thesaurus-based semantic net for knowledgebased search. In *Proceedings of the 12th Innovative Applications of Artificial Intelligence (IAAI'2000)*, pages 988–995. AAAI Press, 2000.

[dC74] N C da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15:497–510, 1974.

[DKT98] P Dung, R Kowalski, and F Toni. Argumentation-theoretic proof procedures for default reasoning. Technical report, Department of Computing, Imperial College, London, 1998.

[DN99] B Duval and P Nicolas. Learning default theories. In Anthony Hunter and Simon Parsons, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 1638 of *Lecture Notes in Computer Science*, pages 148–159, 1999.

[Dun95] P Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[EGH95] M Elvang-Gøransson and A Hunter. Argumentative logics: Reasoning from classically inconsistent information. *Data and Knowledge Engineering*, 16:125–145, 1995.

[EGKF93] M Elvang-Gøransson, P Krause, and J Fox. Dialectic reasoning with classically inconsistent information. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, pages 114–121. Morgan Kaufmann, 1993.

[ES01] M Erdmann and R Studer. How to structure and access XML documents with ontologies. *Data and Knowledge Engineering*, 36:317–335, 2001.

[FD00] J Fox and S Das. *Safe and Sound: Artificial Intelligence in Hazardous Applications*. MIT Press, 2000.

[Fis88] A Fisher. *The Logic of Real Arguments*. Cambridge University Press, 1988.

[FJR98] J Fox, N Johns, and A Rahmanzadeh. Disseminating medical knowledge: The PROforma approach. *Artificial Intelligence in Medicine*, 14:157–182, 1998.

[FKA92] J Fox, P Krause, and S Ambler. Arguments, contradictions, and practical reasoning. In *Proceedings of the 10th European Conference on Artificial Intelligence*, pages 623–627. John Wiley, 1992.

[FKEG93] J Fox, P Krause, and M Elvang-Gøransson. Argumentation as a general framework for uncertain reasoning. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1993.

[Gef96] H Geffner. A formal framework for causal modelling and argumentation. In D Gabbay and H-J Ohlbach, editors, *Practical Reasoning*, number 1085 in Lecture Notes in Computer Science. Springer, 1996.

[GHR94] D Gabbay, I Hodkinson, and M Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.

[GK97] T.F. Gordon and N. Karaçapilidis. The Zeno argumentation framework. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 10–18, New York, 1997. ACM Press.

[GK98] J Gebhardt and R Kruse. Background and perspectives of possibilistic graphical models. In A Hunter and S Parsons, editors, *Applications of Uncertainty Formalisms*, Lecture Notes in Computer Science. Springer, 1998.

[GLV91] D Gabbay, E Laenens, and D Vermeir. Credulous vs. sceptical semantics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*. Morgan Kaufmann, 1991.

[Gri97] R Grishman. Information extraction techniques and challenges. In M Pazienza, editor, *Information extraction*. Springer, 1997.

[Hae98] R Haenni. Modelling uncertainty with propositional assumptions-based systems. In A Hunter and S Parsons, editors, *Applications of Uncertainty Formalisms*, pages 446–470. Springer, 1998.

[Hag97] J Hage. *Reasoning with Rules: An Essay on Legal Reasoning and Its Underlying Logic*. Kluwer, 1997.

[Hes99] A Hessami. Risk: A missed opportunity. *Risk and Continuity Journal*, 2:17–26, 1999.

[HH01] A Hessami and A Hunter. Formalisation of weighted factor analysis. *Knowledge-Based Systems*, 2001. (in press).

[HKL00] R Haenni, J Kohlas, and N Lehmann. Probabilistic argumentation systems. In D Gabbay and Ph Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5*, pages 221–288. Kluwer, 2000.

[HM98] A Hunter and P McBrien. Default databases: Extending the approach of deductive databases using default logic. *Data and Knowledge Engineering*, 26:135–160, 1998.

[Hun00a] A Hunter. Merging inconsistent items of structured text. *Data and Knowledge Engineering*, 34:305–332, 2000.

[Hun00b] A Hunter. Ramification analysis using causal mapping. *Data and Knowledge Engineering*, 32:1–27, 2000.

[Hun00c] A Hunter. Reasoning with contradictory information using quasi-classical logic. *Journal of Logic and Computation*, 10:677–703, 2000.

[Hun00d] A Hunter. Reasoning with inconsistency in structured text. *Knowledge Engineering Review*, 15:317–337, 2000.

[Hun01a] A Hunter. First-order quasi-classical logic. In S Benferhat and Ph Besnard, editors, *Qualitative and Quantitative Approaches to Reasoning with Uncertainty*, volume 2143 of *Lecture Notes in Computer Science*, pages 544–556. Springer, 2001.

[Hun01b] A Hunter. Merging structured text using temporal knowledge. *Data and Knowledge Engineering*, 2001. (in press).

[Hun01c] A Hunter. Ramification analysis with structured news reports using temporal argumentation. Technical report, Department of Computer Science, University College London, 2001.

[Jen96] F Jenssen. *An Introduction to Bayesian Networks*. UCL Press, 1996.

[Jen01] F Jenssen. *Bayesian Networks and Decision Diagrams*. Springer, 2001.

[JV00] H. Jakobovits and D. Vermeir. Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, 9:215–261, 2000.

[KAEF95a] P Krause, S Ambler, M Elvang-Gorannson, and J Fox. Is there a role for qualitative risk assessment. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995.

[KAEF95b] P Krause, S Ambler, M Elvang-Gorannson, and J Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11:113–131, 1995.

[KKT93] A Kakas, R Kowalski, and F Toni. Abductive logic programming. *Journal of Logic and Computation*, 2:719–707, 1993.

[KW87] J De Kleer and B Williams. Diagnosing mulitple faults. *Artificial Intelligence*, 32:97–130, 1987.

[LNA+97] R.P. Loui, J. Norman, J. Alpeter, D. Pinkard, D. Craven, J. Linsday, and M. Foltz. Progress on Room 5: A testbed for public interactive semi-formal legal argumentation. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 207–214, New York, 1997. ACM Press.

[LV90] E Laenens and D Vermeir. A fixpoint semantics for ordered logic. *Journal of Logic and Computation*, 1:159–185, 1990.

[Mil95] G Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[MP99] P McBurney and S Parsons. Truth or consequences: Using argumentation to reason about risk. Technical report, Department of Computer Science, University of Liverpool, 1999.

[MP01] P McBurney and S Parsons. Representing epistemic uncertainty by means of dialectical argumentation. *Annals of Mathematics and Artificial Intelligence*, 32:125–169, 2001.

[MR70] R Manor and N Rescher. On inferences from inconsistent information. *Theory and Decision*, 1:179–219, 1970.

[MT88] W Mann and S Thompson. Rhetorical structure theory: Towards a function theory of text organization. *Text*, 3:243–281, 1988.

[Nie95] I Niemela. Towards efficient default reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 312–318, 1995.

[Nut88] D Nute. Defeasible reasoning and decision support systems. *Decision Support Systems*, 4:97–110, 1988.

[Par96] S Parsons. Defining normative systems for qualitative argumentation. In D Gabbay and H-J Ohlbach, editors, *Practical Reasoning*, volume 1085 of *Lecture Notes in Computer Science*, 1996.

[Par98] S Parsons. A proof theoretic approach to qualitative probabilistic reasoning. *International Journal of Approximate Reasoning*, 19:265–297, 1998.

[Par01] S Parsons. *Qualitative approaches to reasoning under uncertainty.* MIT Press, 2001.

[PL92] G Pinkas and R Loui. Reasoning from inconsistency: A taxonomy of principles for resolving conflict. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*. Morgan Kaufmann, 1992.

[Pol87] J Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.

[Pol92] J Pollock. How to reason defeasibly. *Artificial Intelligence*, 57:1–42, 1992.

[Poo88] D Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[Pra97] H Prakken. *Logical Tools for Modelling Legal Argument.* Kluwer, 1997.

[PS97] H Prakken and G Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, 1997.

[PSJ98] S Parsons, C Sierra, and N Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8:261–292, 1998.

[PV00] H Prakken and G Vreeswijk. Logical systems for defeasible argumentation. In D Gabbay, editor, *Handbook of Philosophical Logic*. Kluwer, 2000.

[Rei80] R Reiter. Default logic. *Artificial Intelligence*, 13:81–132, 1980.

[Rei87] R Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[Sch98] T Schaub. *The Automation of Reasoning with Incomplete Information: From Semantic Foundation to Efficient Computation.* Springer, 1998.

[Sil96] J Sillince. Would electronic argumentation improve your ability to express yourself. In B Gorayska and J May, editors, *Cognitive Technology: In search of a Humane Interface*, pages 375–387. Elsevier, 1996.

[Sil97] J Sillince. Intelligent argumentation systems: Requirements, models, research agenda, and applications. In A Kent, editor, *Encyclopedia of Library and Information Science*, pages 176–217. Marcel Dekker, 1997.

[SL92] G Simari and R Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.

[Tou58] S Toulmin. *The Use of Argument.* Cambridge University Press, 1958.

[Ver95] B Verheij. Arguments and defeat in argument-based nonmonotonic reasoning. In C Pinto-Ferreira and N Mamede, editors, *Progress in Artificial Intelligence: 7th Portugese Conference on Artificial Intelligence (EPIA'95)*, volume 990 of *Lecture Notes in Computer Science*, pages 213–224. Springer, 1995.

[Ver99] B Verheij. Automated argument assistance for lawyers. In *Proceedings of the International Conference on Artificial Intelligence and Law.* ACM Press, 1999.

[vGK87] F van Eemeren, R Grootendorst, and T Kruiger. *Handbook of Argumentation Theory.* Foris, 1987.

[VHvdH98] B. Verheij, J.C. Hage, and H.J. van der Herik. An integrated view on rules and principles. *Artificial Intelligence and Law*, 6:3–26, 1998.

[VLG98] D Vermeir, E Laenens, and P Geerts. Defeasible logics. In *Handbook of Defeasible Reasoning and Uncertainty Management*, volume 2. Kluwer, 1998.

[Vre97] G Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90:225–279, 1997.

[Wag91] G Wagner. Ex contradictione nihil sequitur. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'91)*, 1991.

[Wal89] D Walton. *Informal Logic: A Handbook for Critical Argumentation*. Cambridge University Press, 1989.

[Wel88] M Wellman. Qualitative probabilistic networks for planning under uncertainty. In J Lemmer and L Kanal, editors, *Uncertainty in Artificial Intelligence 2*. Elsevier, 1988.

[Wel90] M Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257–304, 1990.

[WSG96] Y Wilks, B Slator, and L Guthrie. *Electric Words: Dictionaries, Computers, and Meanings*. MIT Press, 1996.