Logical Models of Argument

CARLOS IVÁN CHESNÉVAR and ANA GABRIELA MAGUITMAN

Universidad Nacional del Sur

AND

RONALD PRESCOTT LOUI

Washington University

Logical models of argument formalize commonsense reasoning while taking process and computation seriously. This survey discusses the main ideas that characterize different logical models of argument. It presents the formal features of a few main approaches to the modeling of argumentation.

We trace the evolution of argumentation from the mid-1980s, when argument systems emerged as an alternative to nonmonotonic formalisms based on classical logic, to the present, as argument is embedded in different complex systems for real-world applications, and allows more formal work to be done in different areas, such as AI and Law, case-based reasoning and negotiation among intelligent agents.

Categories and Subject Descriptors: I.2.4 [Computing Methodologies]: Artificial Intelligence—Knowledge representation formalisms and methods

General Terms: Reasoning, Argumentation

Additional Key Words and Phrases: defeasible argumentation, argumentative systems, defeasible reasoning

1. MODELING ARGUMENT IN ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) has long dealt with the challenge of modeling commonsense reasoning, which almost always occurs in the face of incomplete and potentially inconsistent information [McCarthy 1984; McCarthy and Hayes 1969]. A logical model of commonsense reasoning demands the formalization of principles and criteria that characterize valid patterns of inference. In this respect, classical logic proved to be inadequate [Reiter 1980], since it behaves monotonically.

Within AI, several nonmonotonic reasoning formalisms emerged to meet this challenge. In these formalisms, conclusions...
drawn may be later withdrawn when additional information is obtained. Formal logics of argument emerged as one style of formalizing nonmonotonic reasoning.

The literature on nonmonotonic reasoning dominated AI's journals in the mid-1980s. Oddly, the discovery of relevant philosophical, legal, and rhetorical traditions coincided with a decline in AI's purely mathematical interest in the subject as well as with a rise of international interest. Modeling argument appears to be at the foundation of AI's understanding of rule-based systems (where rules can come into conflict). It also differentiates AI's stake in representation languages (what is called knowledge representation and reasoning) from philosophical logic. At times, AI chooses logics for representation (for specification), in the same way that programmers choose programming languages. At other times, AI uses logic for analysis (for example, the way it is used in algorithms analysis). Argumentation clearly serves the former, representational use of logic in AI. AI has a particular interest in nonstandard logics. Those logics have greater expressive power, especially in a practical sense, or correspond more naturally to human linguistic convention and inferential limitation.

When a rule supporting a conclusion may be defeated by new information, it is said that such reasoning is defeasible [Nute 1988a; Pollock 1974]. When we chain defeasible reasons to reach a conclusion, we have arguments, instead of proofs. It makes sense to require defeasible reasons for argumentation. Arguments may compete, rebutting each other, so a process of argumentation is a natural result of the search for arguments. Adjudication of competing arguments must be performed, comparing arguments in order to determine what beliefs are justified. Since we arrive at conclusions by building defeasible arguments, and since mathematical argumentation has so often called itself argumentation, we sometimes call this kind of reasoning defeasible argumentation.

At first, an argument-based approach to modeling commonsense reasoning drove the development of new logical languages, resulting in new formalisms, which extended classical logic for performing nonmonotonic reasoning. At this time, argument was energetically compared with logics of counterfactual conditionals (conditional logic) and with induction.

Recently, defeasible argumentation has looked more broadly to patterns of reasoning recognized outside of mathematical logic: law, political science, rhetoric, and even scientific arguments. The result has been to recast AI's original nonmonotonic reasoning into something that pays more attention to computation.

The well-known example of an argument from AI is Tweety flies because Tweety is a bird, which can be counterargued by the argument But Tweety is different, so perhaps Tweety does not fly. In epistemology, the standard example is This looks red, therefore it is red, which can be counterargued by But the ambient light is red, so perhaps it is not red. In law, the famous example is A contract exists because there was offer, acceptance, memorandum, and consideration, which can be counterargued by But one of the parties to the contract is incompetent, so there is no contract. These are one-step arguments in a two- ply dialogue. Arguments can chain and dialogues can run deeper.

This paper presents the main approaches to logical models of argument. The paper is structured as follows: first, in Section 2., we discuss the main ideas that motivated the development of argumentation frameworks for modeling different kinds of commonsense reasoning. We also discuss the development of applications based on argumentation. Next, in Section 3, we analyze the most relevant technical features of prominent formalisms for argumentation. Finally, in Section 4, we discuss the conclusions that have been obtained.

2. BACKGROUND

The use of arguments and defeasibility has a long tradition in Western history, being rooted in the Aristotelian tradition of practical reasoning, in ethics, and in
law. In the last decades, researchers from a wide range of fields (such as law, philosophy, decision making, and computer science) have made important contributions in developing models of argumentation.

In the philosophy of law, defeasibility was a well-known idea and technical term before it emerged in other areas. In 1958, Stephen Toulmin [Toulmin 1958] introduced a conceptual model of argumentation. He considered a pictorial representation for legal arguments, in which four parts are distinguished: claim, warrant (a non-demonstrative reason that allows the claim), datum (the evidence needed for using the warrant), and backing (the grounds underlying the reason). Counterarguments are also arguments that may attack any of the preceding four elements. By chaining diagrams a disputation can be visualized. Today, Toulmin’s work is essentially of historical interest. It suggested that the proper formalism for argumentation was to be found in deontic logic, which was fashionable at that time.

H. L. A. Hart’s “Ascription of Responsibility and Rights” [Hart 1951] is where we find perhaps the first clear pronouncement of defeasibility and the technical introduction of the term. Despite multiple criticisms by philosophers of ethics and philosophers of language outside of law, there are many reasons to believe that Hart’s view was valid and prescient.

In the philosophy of logic, several traditions are closely related to argumentation and even contributed formal ideas to the study of argumentation. Intuitionist and paraconsistent logic, conditional logic, belief revision, and dialogue logic deserve mention. Lately, deontic logic has embraced a defeasible conditional, and deontic concerns of obligation and permission are a natural match for argumentation. Some families that have not been related to (and should not be confused with) argumentation are fuzzy logic, logics of high probability, induction, and abduction.

John Pollock [1974] and others brought the idea of defeasibility into epistemology to address questions of justification. Later Pollock integrated his philosophical work with computer science, developing a theory of defeasible reasoning [Pollock 1987, 1991a]. Pollock postulates that reasoning operates in terms of reasons. Reasons can be assembled to comprise arguments. He distinguishes two kinds of reasons, nondefeasible and defeasible. The notion of defeasible reason, also known as prima facie reason, is defined in terms of a special kind of knowledge called defeaters. Defeaters are new reasons that attack the justificatory power that a certain (defeasible) reason has on behalf of its conclusion. Pollock refers to two kinds of defeaters, rebutting defeaters and undercutting defeaters. A rebutting defeater is a reason that attacks a conclusion by supporting the opposite one, while an undercutting defeater is a reason that attacks the connection existing between a reason and a conclusion. Pollock presents also an analysis of the notion of warrant. For a conclusion to be warranted it is not enough to have an argument for it. A proposition is warranted if it emerges undefeated from an iterative justificatory process (not to be confused with a search process) in which the same argument can be successively defeated and reinstated.

It is well known that AI has received contributions from many fields, and argumentation is not an exception in this sense. Within the AI community, however, the need of some model of argument for commonsense reasoning can be traced to Jon Doyle’s work on truth maintenance systems [Doyle 1979]. Doyle offered a method for representing beliefs together with the justifications for such beliefs, as well as procedures for dealing with the incorporation of new information. Later, Doyle sketched the computational structures of a program intended to model a conscious, adaptive intelligent agent, and considered the role of defeasibility in theory formation and decision making.

\[^{1}\text{This was applied later for user-interface applications in legal frameworks (see Marshall et al. [1991])}.\]

\[^{2}\text{Actually, Chisholm and Sosa made considerable use of defeasibility, but without technical sophistication.}\]
C. I. Chesñevar et al.

[Doyle 1980]. In his system, reasons are defeasible. As he points out [Doyle 1980];

after an inference has been made, it can be reflected on. If reflection determines that the reason was mistaken because, for example, the inference was made in exceptional or special-case circumstances in which it was not strictly valid, the program can defeat the reason by providing a defeating reason. This defeating reason may in turn be defeated by other reasons. The defeasibility of reasons allows the program to change any of its attitudes, for each attitude is held only because of some reason, and can be rejected by defeating all of its reasons.

Doyle’s truth maintenance system had undeniable affinities to argument that later nonmonotonic logics did not.

At the same time, Raymond Reiter developed his default logic [Reiter 1980] and wrote a famous paper summarizing the main systems advanced so far in AI to cope with the problem of nonmonotonicity [Reiter 1980]. In the early 1980s, several logical frameworks were proposed as valid approaches for modeling commonsense reasoning, with no major success. Argument based systems did not exist in those years of pioneering work in AI, until the mid-1980s, when some epistemologists and philosophers became interested in the subject. The contributions of John Pollock and Donald Nute played a major role in linking epistemology and AI. Ever since, there has been an increasing cross-breeding between these disciplines, which has proved to be fruitful particularly in this decade.

Next we will discuss some important contributions in major areas of argumentation. First, in Section 2.1, we consider the role of philosophy of logic in the evolution of argumentation. We also discuss some logical formalizations which deserve special attention. In Section 2.2 we consider argumentation as a dialectical process, a viewpoint that was introduced by the philosopher Nicholas Rescher in the mid-1970s, brought into life in AI more than a decade later. The evolution of criteria for argument comparison, which constitute a central piece in any argumentative framework, is discussed in Section 2.3. Finally, in Section 2.4, we mention some significant application areas of argument-based systems.

2.1 Argument as a Kind of Logic

Outside the philosophy of law, early work of the philosophical logicians deserves special mention. Kuno Lorenz’s strip diagrams [Lorenz 1961] paved the way for Paul Lorenzen and Jaakko Hintikka’s game-theoretic semantics and Jim MacKenzie’s dialogue systems [MacKenzie 1981]. All of these have stronger links to nondefeasible forms of reasoning than to defeasible forms. Therefore, they are closer to the mainstream of logic. They do, however, presage the dialogical aspects of argumentation and they are the proper bridge between argument and belief revision.

One member of this community whose work is instructive is Carlos Alchourrón [1993], because he studied the relation between defeasibility, deontic logic, and belief revision. In fact, he did not support defeasible rules as the proper way of expressing incomplete knowledge. Instead, he suggested revising the material conditionals, as revision becomes necessary, whenever conflict arises. However, he was among those who understood and appreciated many aspects and implications of defeasibility in logic and law (e.g., its ampliative character).

Logic experienced a huge development in the last century, particularly as a result of research on the foundations of mathematics, and attempts to formalize it as being sound. This is one of the reasons why the first stage in mathematical work on argument did not consider process; instead, it attempted to provide a deviant conditional logic on a nonstandard theory of entailment, even when defeasible conditionals became commonplace. Early work on nonmonotonicity can be considered work on argumentation so long as the work did not try too hard to force “the logical straitjacket” [Cheeseman 1985].

The first steps out of the logical mold included the works of the philosophers Ernest Adams [Adams 1975] (who studied
the logic of conditionals) and Clark Glymour and Rich Thomason, who tried to relate nonmonotonicity to belief revision in an early paper [Glymour and Thomason 1984]. Later work that stayed close to logic focused on explanations of defeasibility in familiar logical terms. Sarit Kraus, Daniel Lehmann, and Menachem Magidor [Kraus et al. 1990] analyzed general patterns of nonmonotonic reasoning, isolating properties of nonmonotonic consequence relations. In their analysis, they characterize these relations using representation theorems, which connect the semantic and the proof-theoretic points of view. Their work paved the way for refining consequence within argumentative frameworks [Vreeswijk 1993]. Héctor Geffner and Judea Pearl [Geffner 1991; Geffner and Pearl 1992] define a preferred-model semantics for any ordered set of defeasible conditionals, as well as a sound and complete proof theory for this semantics. Salem Benferhat, Didier Dubois, and Henri Prade [Benferhat et al. 1992; 1997] considered the representation of defaults in possibilistic logic.

In 1985, the philosophical logician Donald Nute produced a "defeasible conditional logic" in which defeat was based on the logical strength of antecedents of rules. In Nute’s system, the competition was among rules, not among arguments (structured collections of rules). Later he also defined a simple and elegant system of defeasible reasoning, called LDR1, which was refined two years later by introducing adjudication among competing arguments based on the specificity of the top-rule [Nute 1988a; 1988b; 1988c].

James Delgrande [1986] created a conditional logic based on the idea of irrelevance. Evidence is irrelevant in Delgrande’s sense if it contributes only to arguments that are defeated. Delgrande’s system found wide acceptance among AI’s logicians, because AI insisted on a semantical understanding of the defeasible conditional. Current work instead follows Nute, Pollock, Reiter, and Doyle, and inductive logic, pushing the defeasible relation into the metalanguage. This was the main point that the logician W. V. O. Quine made when asked about defeasibility (see Simari and Loui [1992]).

Fangzhen Lin and Yoav Shoham [Lin and Shoham 1989] developed an abstract argument system together with methods aimed at reformulating some well-known systems of default reasoning in terms of their framework [Lin and Shoham 1989]. They show how the framework can be applied to formalize inheritance systems, temporal projection and temporal diagnosis. The system is very general and many issues remain unspecifed; for example, there is no notion of preference between conflicting arguments. However, they introduced some notions (e.g., the definition of argument as a proof tree, although Loui [1987] had already defined arguments as graphs) which proved to be useful in other, more evolved systems.

Robert L. Causey devised EVID [Causey 1990], a system for interactive practical defeasible reasoning. Application programs using this system can infer a conclusion defeasibly from a conjunction of supporting facts, along with an appropriate general rule. This particular inference of the conclusion might be defeated by additional facts, although other independent evidence could still support the conclusion on the basis of other rules. EVID was implemented in PROLOG and included an extensive logical interface that permits the user to interact with and override an application program’s defeasible conclusions.

There were many early attempts at bridging the gap between logic programming and argumentation. Matthew Ginsberg and Andy Baker worked on theorem proving using prioritized circumscription [Ginsberg and Baker 1989]. The work of Phan Minh Dung [1995b] is particularly relevant in bridging the gap between logic programming and argumentation. Within first-order logic, Dung presents a theory for argumentation whose central notion is the acceptability of arguments. He proves that argumentation can be viewed as a special form
of logic programming with negation as failure, introducing thus a general logic-programming-based method for generating meta-interpreters for argumentation systems. Later, Bondarenko et al. [1997] developed an abstract argumentation-theoretic framework, which incorporates many of Dung’s ideas.

Guillermo Simari’s Ph.D. Thesis [Simari 1989; Simari and Loui 1992] presents a system for defeasible reasoning in which a general theory of warrant (following Pollock [1987]) and the definition of specificity (as proposed by Poole [1988]) are combined. His main concern is to characterize the conditions under which an argument structure is preferred. The resulting system, generally referred to as the Simari–Loui framework, evolved so that dialectical considerations were taken into account [Simari et al. 1994b].

The Simari–Loui framework was used as a basis for many different works. Marcelo Falappa [Falappa 1999; Falappa and Simari 1995; Simari and Falappa 1995] introduces a belief revision system [Alchourrón et al. 1985] using a special kind of argument. He considered revising a database using arguments, thus developing a semirevision operator. In his approach, new beliefs forcing revision do not have absolute preeminence over the agent’s existing beliefs. Hence, when the agent receives a new belief a, such that a is in contradiction with its knowledge K (i.e., K ⊨ a), an argument for a is required. Should this argument be stronger than any possible argument against it, then a is accepted as a new belief. Gustavo Bodanza [Bodanza and Simari 1995] analyzed defeasible reasoning in disjunctive databases. Claudio Delrieux [1995] introduces a system of plausible reasoning. The combination of this system with Simari–Loui’s framework gives rise to a richer system in which a plausibility order among information sources induces a plausibility order among arguments. Juan Carlos Augusto studied the construction of arguments for temporal reasoning. He extended Simari–Loui’s framework to explicitly handle temporal information [Augusto 1998; Augusto and Simari 1994] and adapted his system for solving planning problems involving time [Augusto 1998; Augusto and Simari 1995].

The work of Ramathan Guha and Douglas Lenat [Guha 1990; Guha and Lenat 1989] deserves mention, although their contribution belongs to application of argument within a logical framework rather than defining argument-based logical inference. They were aware of the importance of argumentation when defining inference within the Cyc project. They considered arguments as an extension of the concept of a proof. Arguments might contain a class of sentences labeled as assumptions. Given those arguments, Cyc compares them and decides what to add to the knowledge base. Arguments are first-class objects in the Cyc knowledge representation language, and they are compared by using axioms in the Cyc knowledge base that capture intuitions of various aspects of default reasoning. Guha’s work was also devoted to considering reasonable preferences and dialectic patterns [Guha 1990].

2.2 Argument and Dialectical Process

An argument produces warrant for its claim in a context of counterarguments, rebuttal, and further counterargument. It produces warrant as the outcome of a process, which subjects the claim to dispute. The approaches to argumentation presented in the previous section paid little attention to this process; instead, argumentation was assumed to take no time, or else warrant was defined with respect to all constructible arguments.

In argumentation, one of the right kinds of process for constructing rational belief is dialectic.5 Dialectic refers to one form

---

4 In Section 3.3 we describe this system in detail.

5 The term dialectic was introduced by Aristotle and can be defined as “that discipline which investigates and formulates the rules for disputuation concerning dialectical problems. Dialectical problems are those for which no apodictic (demonstrable, incontrovertible) knowledge is available; instead, there are assumptions which are acceptable or plausible to solve them.” [Hegselmann 1985]
of disputation in which a serializable resource is distributed so that one party's use of that resource is informed by the result of the other party's (or parties') prior use of resource. The serialized resource is typically either a search for arguments or time for presentation of arguments [Loui 1993b].

The role of characterizing the process that is needed for constructing rational belief remained quite unseen in the early formalizations of argument in AI. Nicholas Rescher's *Dialectics* [1977], which followed his monograph on plausible reasoning, is one of the most relevant philosophical works on dialectic-based argumentation.6 He develops a formal system intended as a disputational approach to the theory of knowledge. His system augments Lorenz strip diagrams mainly by adding nondemonstrative reasons and informally discussing termination based on plausibility.

In philosophy, Rainer Hegselmann [1985] analyzed the major steps leading to a proper formalization of dialectic, by discussing the methods, heuristics and techniques to be applied. He also formalized a *Grundmodell* (basic model) for dialectical reasoning. He prefers the term *formal dialectics* rather than *dialectics*, associating the latter with Aristotle's original proposal, which lacks formal tools.

Most of John Pollock's work on defeasible reasoning in the last two decades has been devoted to investigating the processes to be performed by an intelligent agent so that his or her conclusions and decisions can be considered as rational. It is possible to conceive of Pollock's processes as dialectical processes, though he is often not explicit on this point. Most of his designs have been embedded into *Oscar*, a fully implemented architecture for rational agents, written in *Lisp*, based upon a general purpose defeasible reasoner. Although the implementation makes use of natural deduction, it is probably dialectical in its use of search, like Baker and Ginsberg's *Lisp* implementation of their system, and probably for the same reasons.

Ronald Loui [1991a; 1993b] explicitly investigated the appropriateness of formal dialectics as a basis for nonmonotonic reasoning and defeasible reasoning that takes computational limits seriously. In Loui's approach, rules that come into conflict should be regarded as policies, which are inputs to deliberative processes. Dialectical protocols are appropriate for such deliberations when resources are bounded and search is serial. Loui's work is oriented also toward capturing reasoning patterns in legal argumentation and conceptualizing *argument games*, which help clarify how resource-bounded reasoning should be performed. Loui and his students implemented his ideas for dialectical protocols and argument games, developing several programs such as *Nathan*, *Sophie*, *Lanop*, and *Antigone*, written in *C*, *Gawk*, *Lisp*, and *C++*.

Gerard Vreeswijk [1993] developed a theory of argumentation called *abstract argumentation systems*.7 Using this theory as a basis, he explored dialectical issues when modeling the procedure of justification as a debate between two parties, proponent and opponent. Vreeswijk implemented *Iacas*, a program written in *Lisp* to do interactive argumentation on a computer [Vreeswijk 1994]. The program is intended to demonstrate Vreeswijk's theory of argumentation. Later, Vreeswijk extended some of his results into novel areas, such as multiagent systems [Vreeswijk 1995b] and meta-games for changing the rules of argument games.

Thomas Gordon [Gordon 1994] formulated *The Pleadings Game* as a normative formalization and computational model of civil pleading, founded in Robert Alexy's discourse theory of legal argumentation. The consequences of arguments and counterarguments are modeled using Geffner and Pearl's *conditional entailment* [Geffner and Pearl 1992]. Conflicts between arguments can be resolved by arguing about the validity and priority of rules. The computational model was fully

---

6We devote Section 3.1 to discuss the main technical issues of Rescher's approach.

7We consider this theory in Section 3.4.
implemented. The Pleadings Game uses the concepts of issue and relevance to focus the discourse. A tractable inference relation is used to commit players to consequences of their claims. The goal of the game is identifying issues, rather than deciding the main claim.

Gerhard Brewka [1994b] has developed a formal reconstruction of Nicholas Rescher’s theory of disputation [Rescher 1977]. His approach incorporates some elements of Reiter’s Default Logic [Reiter 1980], where Rescher’s provisoed assertions are seen as Reiter’s defaults. The notion of extension is also borrowed from Reiter’s approach, and it is modified in order to define SDL extensions. SDL extensions are default extensions that deal with specificity.

Guillermo Simari, Carlos Chesñevar, and Alejandro García [Simari et al. 1994b] considered the role of dialectics in defeasible argumentation. They considered some of Rescher’s ideas within the Simari–Loui framework [Simari and Loui 1992]. The resulting, evolved framework, called MTDR, is able to cope with fallacious argumentation. Using MTDR as a basis, García developed a knowledge representation formalism for defeasible logic programming and implemented a logic programming language for dealing with defeasible rules. Chesñevar [1996] analyzed strategies for speeding up the inference process by pruning the search space for arguments to determine whether an argument supports justified belief.

Henry Prakken and Giovanni Sartor [Prakken and Sartor 1996b] study some logical notions of legal arguments by presenting a system of defeasible argumentation or argumentation framework. They contend that for a theory of defeasible argumentation to be realistic it should also formalize arguments about priorities. This gives rise to their second argumentation framework, in which priorities are not fixed but defeasible. Prakken [1998] also developed a dialectical proof theory for those logical systems for defeasible argumentation that fit the abstract format developed by Dung, Kowalski, and others [Bondarenko et al. 1993]. Prakken shows how his proof theory can also handle reasoning about preference criteria for comparing competing arguments.

Bart Verheij’s approach [Verheij 1996] to argumentation is twofold. On the one hand, he analyzes the role of rules and reasons in argumentation, their relevant properties, and characteristics. On the other hand, he considers the role of process in argumentation, analyzing what determines that a given argument should be defeated in a particular argumentation stage. Verheij studies argumentation and defeat from these two angles, resulting in formalisms of different nature, RBL (reason-based logic) and CUMULA, reason-based logic [Hage and Verheij 1995], introduced by Jaap Hage [1993], gives a model of the nature of rules and reasons in which the properties of rules and reasons can be investigated. CUMULA is a model of argumentation in stages that provides a framework for analyzing issues such as how the structure of an argument is related to defeat, when arguments are defeated by counterarguments, and how the status of arguments is affected by the argumentation stage. Arno Lodder [1998] just completed his dialectical model of argumentation, which is a mix of the ideas from RBL, dialogue logic, and Juergen Habermas’s consensus theory of truth. These authors are considered again in the later section on applications to legal reasoning.

Robert Kowalski and Francesca Toni [1996] have outlined a formal theory of argumentation, in which they argue that defeasible reasoning with rules of the form $P \text{ if } Q$ can be understood as “exact” reasoning with rules of the form $P \text{ if } Q \text{ and } S \text{ cannot be shown}$, where $S$ stands for one or more defeasible “nonprovability claims.” With this understanding of defeasibility, argumentation turns out to be a dialectical process in which proponent and

---

8Brewka’s approach is described in detail in Section 3.5.
8Some features of MTDR are discussed in Section 3.3.
10See Section 3.7 for details.
11See Section 3.8 for more details.
opponent exchange “exact” arguments, attacking previous nonprovability claims. Their approach allows them to reduce all forms of defeasibility to that of nonprovability claims. Rebuttal and priority attacks can be reformulated as undermining nonprovability claims. Admissibility semantics [Bondarenko et al. 1997] can also be elegantly defined within their framework. Because their approach is abstract, it can be formalized in many formalisms, such as default logic, extended logic programming, and nonmonotonic modal logic.

2.3 Argument Comparison

The work on adjudicating conflict when reasoning with incomplete information can be traced back to the “clash of intuitions” [Touretzky et al. 1987] emerging from the existence of too many alternative inheritance systems. The work on mathematical inheritance [Sandewall 1986; Touretzky et al. 1987] was an early step toward comparing arguments. Independently, Lynn Stein [1992] and David Makinson and Karl Schlechta [1991] analyzed two difficulties in the “directly skeptical” approach to inference in defeasible inheritance nets [Touretzky et al. 1987]. They pointed out the problems emerging from floating conclusions and zombie paths [Loui and Dorosh 1991].

Loui presented a formal idea of “defeat among arguments” [Loui 1987] that tried to apply Henry Kyburg’s ideas for probabilistic and inductive inference [Kyburg 1961] to collections of Nute-like defeasible rules [Nute 1988b]. The comparison of arguments referred to the arguments’ structure. There were various criteria for defeat, such as “rule specificity,” “more directness,” “more evidence,” and “preferred premises.” When two different criteria conflicted, there was inconclusive “interference.”

Particularly noteworthy was the work of David Poole [Poole 1985; 1988; Poole et al. 1985], whose framework THEORIST allowed conflicting “scenarios” to be compared using specificity as a preference criterion. His work takes its philosophical background from Karl Popper’s idea of scientific theory formation [Popper 1934; 1979]. Poole offered a new approach for solving the problem of multiple extensions in nonmonotonic reasoning. Poole’s formalism conceptualizes default rules in a different way than predecessors, which incorporate them as an “extension” to classical logic. Poole contends that defaults are possible hypotheses for explaining answers within a given theory, which can be used as long as they are consistent with the rest of the knowledge available. Specificity emerges as a tool for comparing conflicting theories (scenarios) using a syntactic preference criterion.

Héctor Geffner and Judea Pearl [Geffner and Pearl 1992] present a proof theory of defeasible conditionals that is based on the idea of comparing arguments. Their theory is derived from a model theoretic semantics with a preferential model structure. A class of orderings reflects the notion of specificity, on models.

Henry Prakken [Prakken 1993] investigates preference when comparing arguments by considering the ways of combining several kinds of defeat. His work is meant to be independent of legal reasoning, in which different preference criteria such as hierarchy, specificity, and time of enactment of a norm have to be taken into account when comparing conflicting arguments.

Bart Verheij [Verheij 1996] considers different types of defeat, extending Pollock’s notions of rebutting and undercutting defeat. He develops a formal model in which an argument can be defeated by sequential weakening, (by chaining so many reasons that a defeasible conclusion no longer holds) and by parallel strengthening (which refers to the case in which the coordination of two arguments defeats one of the two arguments). More general kinds of defeat occur when there is a group of arguments whose conclusions cannot hold simultaneously, leading to collective and indeterministic defeat.12

12Veheij’s approach is treated in more detail in Section 3.8.
2.4 Arguments: Applications

In this section, we will discuss some important argument-based applications. First we will present some early research results in which argumentation was linked to other areas of computer science, such as modeling natural language. Then we will consider the most important contributions of argument in AI and law, a discipline that has emerged as a rapidly growing field, and where argument is called upon to play a major role. Finally, we will discuss different kinds of hypertext-based environments which make use of argumentation.

2.4.1 Argument in Natural Language. The work of Sergio Alvarado and Michael Dyer [1985a; 1985b] and L. Birnbaum, M. Flowers, and R. McGuire [Birnbaum et al. 1980] focused on editorial text comprehension using an argumentative framework and can be considered the earliest investigations of argumentation in AI.

Flowers et al. [1982] and Birnbaum [1982] investigated the representation of argument structures for understanding utterances. They considered arguments as networks of propositions connected by support or attack relations. They analyzed argument structure by identifying commonly occurring patterns of support and attack that encompassed several propositions. They used argument molecules as basic entities to specify which propositions (among those contained in the molecule) were worth trying to attack or support. Thus argument molecules could be used to help plan rebuttals, and to generate expectations about an opponent’s possible rebuttals. They concluded that arguments possess useful structural properties, abstracted from the specific propositions they encompass.

Alvarado [Alvarado and Dyer 1985a] postulates the existence of argument units as basic constructs of argument knowledge, which consists of configurations of attack and support relationships related to abstract goal and plan situations. Argument units allow a language comprehension system to recognize and interpret arguments in disparate domains. His work aims at modeling how a refutation or accusation is organized and how this affects the process of comprehension, memory construction, and question answering. His approach manages many different knowledge sources, including scripts, goals, plans, actions, beliefs, and so on. Alvarado implemented a prototype computer program (called OpEd), capable of reading editorial segments in the domain of politico-economics, and answering questions about their argument content.

Martha Pollack and Kurt Konolige [Konolige and Pollack 1989, 1993] made use of an argumentative background to capture interesting properties of the theory of intention. They approached cognitive attitudes of belief and knowledge by providing a representationalist model of intention. The resulting formalism is useful for tasks such as plan recognition, in which one agent must determine the mental state of another agent using partial information. This model of intention would replace the traditional normal modal logics (in which an agent believes all the consequence of his beliefs).

2.4.2 Argumentation in AI and Law. Much of the most recent work in argumentation has been done in the field of artificial intelligence and law. Among the early contributions in this field, the works of Thomas Gordon and Henry Prakken deserve special mention for their early formal work.


Logical Models of Argument

through dialogue games. His most recent work has been oriented toward defining ontologies in legal information systems [Bench-Capon and Visser 1997].

David Skalak and Edwina Rissland [Rissland et al. 1993; Skalak and Rissland 1992] developed several systems for dealing with legal argumentation. They focused particularly on indexing schemes that facilitate argument creation. One of their recent programs is BANKXX, a case-based legal argument program that retrieves cases and other legal knowledge pertinent to a legal argument through a combination of heuristic search and knowledge-based indexing. BANKXX generates arguments by performing a heuristic search on a highly interconnected network of legal knowledge. The legal knowledge includes cases represented from a variety of points of view: cases as collections of facts, cases as dimensionally analyzed fact situations, cases as bundles of citations, and cases as prototypical factual scripts, as well as legal theories represented in terms of domain factors.

As noted earlier, Jaap Hage [Hage 1993; Hage and Verheij 1995] developed a theory about legal reasoning called reason-based logic (RBL), that takes the notion of a reason to be central. His work is inspired by typical legal forms of reasoning such as analogous rule application, weighing reasons, and reasoning about the validity and the application of rules. The basic idea of RBL is that the application of a rule only leads to a reason which pleads for the rule’s conclusion. When arguing for a conclusion, the reasons that plead for and against the conclusion are collected and then weighed to determine which ones prevail. Hage’s work also addresses the topic of metalevel reasoning, considering the use of rules in concrete cases. Next, Hage [1995] incorporated special facilities in RBL for teleological reasoning. RBL served as a theoretical basis for Verheij’s work [Verheij 1996].

Vincent Aleven and Kevin Ashley [Aleven and Ashley 1997; Ashley 1990] designed an intelligent learning environment called CATO, intended to help beginning law students learn basic skills in case-based argumentation. CATO models ways in which experts compare and contrast cases, assess the significance of similarities and differences between cases in light of general domain knowledge, and use the same general knowledge to organize multcase arguments by issues. CATO communicates its model to students by presenting dynamically generated argumentation examples and making visible the argument structure.

The HELIC system, created by K. Nitta et al. [1995], explores knowledge representation issues in AI and law. HELIC is a software tool for legal reasoning that includes an “argumentation function” and a “debating function,” used for building arguments and comparing them, respectively. Nitta’s approach follows Prakken’s [1993] and Sartor’s [1993; 1994]. HELIC takes viewpoints (priorities) as part of the input. For the interpretation of penal rules, they believe that flexibility is more important than strict interpretation. However, viewpoints are unconditional facts, whereas in Prakken’s system [Prakken 1998] they can be conditional upon other facts.

Arno Lodder and Aimée Herzog [Lodder and Herzog 1995] developed a formal elaboration of the theory of legal reasoning and argumentation, based on Jaap Hage’s reason-based logic [Hage and Verheij 1995]. Their aim is to model legal reasoning as a dialogue. Their motivation for this approach is that dialogues fit nicely with legal practice, making it easy to take burden of proof into account. Besides, they consider the nature of law as purely procedural, since there is no law except as the result of applying legal rules to concrete cases. They developed a framework called Dialaw, in which they formalize allowed moves in legal discourse. The framework is intended to assist in analyzing legal decisions and in constructing a rational justification for a solution of a legal conflict. Their framework also allows dealing

13The notion of rule is used in a broad sense, including legal principles.
with the exclusion of rules and the weighing of reasons. Besides, both case-based and rule-based reasoning can be dealt with.

Loui with Jeff Norman defined different criteria for formalizing rationales within an argumentative framework [Loui and Norman 1995]. A rationale for a rule is a structure that contains relevant information about the reason for the rule's adoption. A rationale for a case, similarly, is relevant additional information about the decision reached. For Loui and Norman, the essential question is what the appropriate forms of rationales are and how they can be brought to bear on argument. Their work aims at devising a richer model for argumentation, in which rationales are included. They contend that dialectical moves force reversion to the forms from which rules were compiled. Hence, the model for argumentation might be made more comprehensive, increasing the repertoire of formally modeled moves.

James Palmer [Palmer 1997] tried for a new paradigm in the design of legal knowledge-based systems. His paradigm is based on a structuralist model of legal merit argument. Legal merit arguments are arguments about the desirability of a particular legal outcome measured in terms of moral responsibility, rights, social welfare, and so on. Palmer contends that, despite the apparent diversity among all those kinds of arguments, there exists a remarkable uniformity, characterized by the stock narratives a lawyer uses when arguing (called argument bites) and the recurring interrelationships of support and opposition that exist between competing arguments. His argument bites are reminiscent of Alvarado's argument units, but applied to legal reasoning.

2.4.3 Argumentation in Hypertext. Argumentation has proven to be a useful tool when modeling inference within interactive environments. In most cases, hypertext allows the user to characterize text as introspective dialogue or debate. Hypertext turns out to be the natural way to order and display argument conflict and support.

Raymond McCall and Gerhard Fischer [McCall and Fischer 1989] created a computer-based design aid called JANUS, which utilizes knowledge to link a graphic construction system to hypertext. The JANUS design environment supports construction (the activity of creating the form of the solution) and argumentation (the activity of reasoning about the problem and its solution). In JANUS argumentation is mostly verbal, but partly graphical. Their work is based on Donald Schon's theory of architectural education, also called "reflection-in-action"; construction corresponds to what Schon calls action, and argumentation corresponds to reflection. JANUS promotes reflection-in-action by providing computer support for argumentation.

EUCLID [Bernstein 1992] is software that supports knowledge structuring tasks. They particularly consider the representation of the structure of arguments in hypertext. AQUANET allows the user to represent information graphically in order to explore its structure. Users can customize knowledge structures for specific tasks. The framework is based on an enriched version of Toulmin's diagrams, and it is designed to support users in creating, storing, editing, and browsing large graphical knowledge structures.

Violetta Cavalli-Sforza and Daniel Suthers [Cavalli-Sforza and Suthers 1994] developed BELVEDERE, an environment for practicing scientific argumentation. It is targeted at helping students think about science and scientific theories from a dialectical perspective. In BELVEDERE, students can view and construct arguments using both text and a specialized graphical interface, which embodies key ideas of scientific argumentation. BELVEDERE's graphical language and its internal representation of arguments are loosely based on Toulmin's model of argument [Toulmin 1958], adapted for scientific theories.

2.4.4 Arguments and Decisions. Katia Sycara [Sycara 1989; 1990] developed
Logical Models of Argument

PERSUADER, a framework for intelligent computer-supported conflict resolution through negotiation/mediation. She advocates persuasive argumentation as a mechanism for group problem solving of agents who are not fully cooperative. Construction of arguments is performed by integrating case-based reasoning, graph search, and approximate estimation of agents' utilities. Kraus, Sycara, and Evenchik [Kraus et al. 1998] is an updated and more formal report embodying this same point of view.

Loui [1989b] outlined a formalism for handling defeasible reasoning about decision. He contends that looking at decision analysis as defeasible reasoning produces a framework in which planning and decision theory can be integrated. He considers that defeasible reasoning about decisions is the natural extension of philosophers' defeasible practical reasoning about action, and the difference is that arguments in a decision-theoretic framework are quantitative, often invoking expected utility calculations. Later, Loui [1994] characterized negotiation in terms of an argument game, contending that reason and argument are often legitimate parts of arbitrated and negotiated settlement. He devised a framework that includes a model of case-based argument (based on the CABARET program, from Ashley, Rissland, and Skalak) and a model of case-based reasoning in negotiated settlement (closely related to the logic underlying Sycara's PERSUADER [Sycara 1989]).

John Pollock has considered the classical planning-theory model in AI from a philosophical perspective [Pollock 1992], comparing it with the traditional theory of practical reasoning. Practical reasoning aims at deciding what actions to perform in light of the goal a rational agent possesses. The decision-theoretic model directs activity by evaluating acts one at a time in terms of their expected utilities. Pollock argues that, with some exceptions, this model gives an inadequate theory of practical reasoning, since plans are sometimes embedded one in another, so that they cannot be selected just by maximizing expected values. He proposes a complex criterion, named coextend-ability, to solve this problem. Pollock's approach examines both models in the light of each other and produces a unified model adequate for the purposes of both disciplines. This new model incorporates some concepts of his theory of defeasible reasoning [Pollock 1974; 1995]. Criteria for adjudicating conflict among competing plans are defined.

John Sillince [Sillince 1994] investigated conflict resolution within a computational framework for argumentation. He analyzed how agents attempt to make claims using tactical rules (such as fairness and commitment). In his framework, arguments are constructed by three knowledge sources: quasi-logic, value transfer, and emotional appeal. Agents may support inconsistent beliefs until another agent is able to attack their beliefs with a strong argument.

Kathleen Freeman and Art Farley [Farley and Freeman 1995; Freeman and Farley 1993; 1996] explored dialectical argumentation as a basis for making decisions in domains where knowledge is incomplete, uncertain, or inconsistent. Their model includes a catalog of argument moves and a set of heuristics for selecting moves and thereby controlling argument generation. They particularly address the role that burden of proof plays in the process of dialectical argumentation and provide an operational definition for various levels of proof.

George Ferguson, James Allen, and Brad Miller [Ferguson et al. 1996] implemented a mixed-initiative planning system for solving routing problems in transportation domains. The unifying model of interaction was implemented as a form of dialogue. Both the system and human are participants in a dialogue; the human knows best the high-level goals, whereas the system manages best the low-level details. Positions are argued for reaching a goal chosen by the user (human).

John Fox et al. [Fox and Parsons 1997; Fox et al. 1993] have worked on argument-based projects for real-world applications,
taking advantage of reasoning methods based on the logic of argumentation. In these projects, decision support is based on argument-based frameworks. Simon Parsons and others. Fox and Parsons [1996] and Parsons [1997] have also worked on methods for representing and reasoning with imperfect information, especially the use of argumentation in the context of medical and biomolecular applications. Recent work [Parsons et al. 1998] has been focused on linking agents and argumentation using multi-context systems [Giunchiglia 1993]. In this approach agents are able to deal with conflicting information, making it possible for two or more agents to engage into dialogues to resolve conflicts between them.

Thomas Gordon and others [Gordon and Karacapilidis 1997; Karacapilidis et al. 1996] have worked on the development of ZENO, which creates advanced support for complex multiparty, multigoal decision making. The ZENO system is essentially intended for being used on the World Wide Web to offer assistance to mediators by providing an issue-based discussion forum or conferencing system. The current prototype version is being evaluated in urban and regional planning [Karacapilidis et al. 1997].

Jintae Lee and others [Lee and Lai 1996] have made significant progress in representing and integrating argument analysis within a complex framework. Lee developed DRL [Lee 1990], a language for representing and managing the qualitative elements of decision making, allowing the user to formulate arguments for supporting claims. Using DRL, a framework for evaluating the expressive adequacy of design rationales was developed. The framework is intended for evaluating alternative designs, each with its own rationale.

3. THE STUDY OF SOME ARGUMENTATION SYSTEMS

In this section we will discuss the main technical ideas that characterize some prominent formalizations of argumentation. Most approaches are presented chronologically as they appeared in the literature. The interested reader is referred to Prakken and Vreeswijk [1999] for a deeper insight in some logics for defeasible argumentation.

First we will consider Nicholas Rescher’s approach to dialectics [Rescher 1977], a contribution of the philosophers of logic that remained unseen by the AI community for many years. Next, In Section 3.2 we will discuss Lin and Shoham’s framework [Lin and Shoham 1989], which constitutes maybe the first widely accepted AI paper on argumentation. Then in Section 3.3 we will present the Guillermo Simari–Ronald Louis framework [Simari and Loui 1992], which gave a formal mathematical treatment to defeasible argumentation.

In 1993, Gerhard Vreeswijk’s and Henry Prakken’s Ph.D. theses [Prakken 1993; Vreeswijk 1993] inaugurate the Dutch school of defeasible argumentation in AI. In Section 3.4 we will consider Vreeswijk’s abstract argumentation system [Vreeswijk 1993], in which the notion of debate is introduced as a way of modeling defeasible argumentation. In this context we will also discuss in Section 3.5 Brewka’s extension of default logic, reconstructing Rescher’s approach [Brewka 1994b].

Section 3.6 analyzes the argumentation theory of Phan Minh Dung [Dung 1995b], which connected the logic programming community with defeasible argumentation. Section 3.7 focuses on the system for defeasible argumentation developed by Henry Prakken and Giovanni Sartor [Prakken and Sartor 1996b], which introduces the notion of defeasible priorities. In Section 3.8 we will analyze Bart Verheij’s system for cumulative argumentation [Verheij 1996], which offers a formal model for studying the role of rules and reasons in legal argumentation. Finally, in Section 3.9 we will present the abstract argumentation framework developed by Bondarenko, Dung, Kowalski, and Toni [1997], which linked results from earlier work on nonmonotonic reasoning with defeasible argumentation.
3.1 Rescher’s Dialectics

In his book *Dialectics: A Controversy-Oriented Approach to the Theory of Knowledge* [Rescher 1977], the philosopher Rescher describes formal disputation as a process that involves three parties: the proponent, the opponent, and the determiner. The disputation proceeds in terms of rules and assertions. The structure of a disputation takes the form of matched responses in which the root is the proponent’s initial thesis. Each move in the disputation process consists of one or more fundamental moves. Three kinds of fundamental moves are distinguished:

- **Categorical** assertions, which take the form !\(P\) and are used to establish that “\(P\) is the case.”
- **Cautious** assertions, which take the form \(\downarrow P\) and are used to maintain that “\(P\) is compatible with all that have been shown.”
- **Provisoed** assertions, which take the form \(P = Q\) and are used to establish that “if \(Q\) is the case then usually \(P\) is the case also.”

The first move of a disputation is always a categorical assertion. The possible countermoves to a categorical assertion of the form !\(P\) are a challenge or cautious denial, which takes the form \(\uparrow \sim P\), or a provisoed denial of the form \(\sim P / Q \& \downarrow Q\) for some suitable \(Q\). The responses to a cautious assertion of the form \(\downarrow P\) can be a categorical counterassertion of the form \(\sim P\), or a provisoed counterassertion \(\sim P / Q \& \downarrow Q\), for some suitable \(Q\). The possible attacks on a provisoed assertion of the form \(\sim P / Q\) are a weak distinction \(\sim P / (Q \& R) \& \downarrow (Q \& R)\), or a strong distinction \(\sim P / (Q \& R) \& \downarrow (Q \& R)\), for some suitable \(R\).

Provisoed assertions are assumed to be always “correct” and cannot be challenged directly. This means that after stating the provisoed assertion \(P / Q\), it is not possible for the adversary to assert \(\sim P / Q\). The only way to attack provisoed assertions is to present another provisoed assertion that is equally or better informed than the first one.

The disputation proceeds until either one of the players accepts the arguments (counterarguments) of the other player. If no more reasonable moves are possible, or if some time limit is exceeded, then the determiner can make a decision, based on the plausibility of the proponent’s assertions which were not conceded by the opponent.

Rescher alludes to some constraints of the dialectic game in terms of his theory. A ! move is only available to the proponent, while a \(\downarrow\) move can only be made by the opponent. He also refers to the progressive nature of the disputation process, and distinguishes some illicit moves (e.g., repetition of moves already made is not allowed). Concessions are performed tacitly (i.e., if the opponent does not challenge explicitly an assertion \(A\) performed by the proponent, it is assumed that the opponent concedes \(A\)).

**Example 3.1** [Brewka 1994b] (*Birds typically fly, penguins typically do not fly, and penguins are birds.*). The following is a well-known example in nonmonotonic reasoning literature:

- *Birds typically fly.*
- *Penguins typically do not fly.*
- *Penguins are birds.*

Determining whether a penguin flies can be modeled in Rescher’s model as follows:

<table>
<thead>
<tr>
<th>Proponent</th>
<th>Opponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>!(\sim Flies)</td>
<td>Flies / Bird &amp; (\uparrow Bird)</td>
</tr>
<tr>
<td>!(\sim (Flies / (Penguin&amp;Bird))) &amp; (\downarrow (Penguin &amp; Bird))</td>
<td>(\uparrow \sim Penguin)</td>
</tr>
</tbody>
</table>

Example 3.2 Consider a debate between a proponent \textit{Pro} and an opponent \textit{Opp} about being allowed to drive a small kart for children in a park. \textit{Pro} contends that the kart can be driven in the park, and \textit{Opp} disagrees, adducing that vehicles are usually not allowed in the park and karts can be considered as vehicles. \textit{Pro} contends that the kart is not a vehicle. \textit{Opp} adduces that a kart is propelled by a motor, and this is a reason for considering it a vehicle. \textit{Pro} argues that the kart is not propelled by a motor; \textit{Opp} counterargues by adducing that the kart is self-propelled, which is a reason to believe it is motor-propelled. \textit{Pro} contends that the kart is self-propelled but not motor-propelled, since it has a sail. \textit{Opp} argues then that having wheels is usually a reason for considering a kart to be a vehicle. \textit{Pro} defends himself by arguing that the kart he has is a toy with wheels, which is reason to believe it is not a vehicle. The different issues in this debate are depicted in Figure 1. However, the debate itself does not settle the dispute about \textit{allowed(kart)}.

Some additional notions involved in the process of disputation are discussed. A main concept is that of the “\textit{burden of proof}.” Two conceptions of burden of proof are distinguished. The first one is the \textit{probative burden of an initiating assertion}. This kind of burden rests on the proponent. The second one is the \textit{evidential burden of further reply in the face of contrary considerations}. It has to do with the burden of making the argumentation process \textit{progressive}, and it shifts from side to side as the process evolves. Another tool Rescher describes is the conception of \textit{presumption}. A presumption is a supposition relative to the known facts and will, in general, be defeasible; that is, it must be assumed until some sufficient reason is presented against it. What makes a presumption better than a random proposition is the notion of \textit{plausibility}. Plausibility is a central mechanism in rational dialectics. The plausibility of a presumption will symbolize how well the presumption fits into the set of accepted knowledge. Rescher’s presentation is semiformal. Arguments are logical propositions, which simplifies the possible relationships among conflicting arguments. Rescher foresaw the notion of default rules (characterized as provisoed assertions), which were reinvented by the nonmonotonic reasoning community. He even noted the ampliative character of resource-bounded, nondemonstrative reasoning, compared to nonampliative paradigms. His work touched many of the areas in which defeasible reasoning was going to be involved for a decade, such as search, nonmonotonicity, and defeat criteria.

3.2 Lin and Shoham’s Argument Framework

Fangzhen Lin and Yoav Shoham present an abstract argument system framework together with methods aimed at reformulating some well-known systems of default reasoning in terms of their framework [Lin and Shoham 1989]. The

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Proponent} & \textbf{Opponent} \\
\hline
(1) falowed(kart) & \sim allowed(kart) / vehicle(kart) \\
& \& \top vehicle(kart) \\
(2) \sim \sim vehicle(kart) & vehicle(kart) / motor\_propelled(kart) \\
& \& \top motor\_propelled(kart) \\
(3) \sim motor\_propelled(kart) & motor\_propelled(kart) / self\_propelled(kart) \\
& \& \top self\_propelled(kart) \\
(4) \sim motor\_propelled(kart) / \sim (self\_propelled(kart) \& has\_sail(kart)) \\
& \& \top has\_wheels(kart) \\
(5) \sim vehicle(kart) / \sim (has\_wheels(kart) \& toy(kart)) \\
& \& \top has\_wheels(kart) \& toy(kart)) \\
\hline
\end{tabular}
\caption{A debate in Rescher’s model (Example 3.2).}
\end{table}
scheme is entirely based on inference rules, which are defined over an unspecified logical language. An inference rule can be a base fact of the form $A$, where $A$ is a wff; it can be a monotonic rule of the form $A_1, \ldots, A_n \rightarrow B$, with $n > 0$ and where the $A_i$ and $B$ are wff; or it can be a nonmonotonic rule of the form $A_1, \ldots, A_n \Rightarrow B$, with $n > 0$ and where the $A_i$ and $B$ are wff. Basic facts represent explicit knowledge, monotonic rules represent deductive knowledge, and nonmonotonic rules represent common sense knowledge.

The three kinds of inference rules are used to build arguments. An argument is a rooted tree with labeled arcs. Arguments must be constructed in such a way that the same formula does not appear more than once. An argument supports a formula if the formula is the root of the tree that comprises the argument.

The system is very general, and many issues remain unspecified (e.g., there is no notion of preference between conflicting arguments). On the other hand, Lin and Shoham introduce an important concept, namely the concept of argument structure.

A set $T$ of arguments is an argument structure if the following conditions are satisfied:

1. If $p$ is a base fact, then $p \in T$.
2. $T$ is closed, i.e., $\forall p \in T$, if $p'$ is a subtree of $p$ then $p' \in T$.
3. $T$ is monotonically closed, i.e., if $p$ is built up from $p_1, \ldots, p_n \in T$ by a monotonic rule, then $p \in T$.
4. $T$ is consistent, i.e., does not contain arguments that support both $\Phi$ and $\neg \Phi$ for any wff $\Phi$.

The above conditions for the definition of an argument structure can be strengthened by the completeness condition. An argument system is complete with respect to $\Phi$ if either it contains an argument that supports $\Phi$ or it contains an argument that supports $\neg \Phi$. The set of wffs supported by an argument structure $T$, denoted $Wff(T)$, is defined as follows:

$$Wff(T) = \{ \Phi | \exists p \in T \text{ such that } p \text{ supports } \Phi \}.$$  

Example 3.3 [Lin and Shoham 1989]. Consider the following set of rules:

$$
\begin{align*}
  r_1: & \text{ true } \\
  r_2: & \text{ penguin(a) } \\
  r_3: & \text{ penguin(a) } \rightarrow \text{ bird(a) } \\
  r_4: & \text{ penguin(a), } \neg ab(\text{penguin(a)}) \rightarrow \neg \text{fly(a) } \\
  r_5: & \text{ bird(a), } \neg ab(\text{bird(a)}) \rightarrow \text{fly(a) } \\
  r_6: & \text{ penguin(a) } \rightarrow ab(\text{bird(a)}) \\
  r_7: & \text{ true } \Rightarrow \neg ab(\text{penguin(a)}) \\
  r_8: & \text{ true } \Rightarrow \neg ab(\text{bird(a)}) \\
\end{align*}
$$

The above set of rules gives rise to the following arguments:

$$
\begin{align*}
  p_1: & \text{ true } \\
  p_2: & \text{ penguin(a) } \\
  p_3: & p_1 \triangleright \neg ab(\text{penguin(a)}) \\
  p_4: & p_1 \triangleright \neg ab(\text{bird(a)}) \\
  p_5: & p_2 \triangleright \neg \text{bird(a) } \\
  p_6: & p_2 \triangleright ab(\text{bird(a)}) \\
  p_7: & p_3, p_2 \triangleright \neg \text{fly(a) } \\
  p_8: & p_4, p_5 \triangleright \text{fly(a) } \\
\end{align*}
$$

Assuming the given rules and arguments we can build the following two argument structures:

$$
\begin{align*}
  T_1 &= \{ p_1, p_2, p_5, p_6 \} \\
  T_2 &= \{ p_1, p_2, p_6, p_3, p_7 \} \\
\end{align*}
$$

In this case we can say that the argument structure $T_2$ is complete with respect to $ab(\text{penguin(a)})$ and $ab(\text{bird(a)})$. The set of wffs supported by the argument structure $T_2$ is

$$Wff(T_2) = \{ \text{true}, \text{penguin(a), bird(a), } \neg ab(\text{penguin(a)}), ab(\text{bird(a)}), \neg \text{fly(a)}) \}.$$ 

The authors show how to use the resulting framework to capture default reasoning, autoepistemic logic, negation as failure, and circumscription. They also show how the framework can be applied to formalize inheritance systems, temporal projection and temporal diagnosis. This apparent power may be due to the simplicity.
of the system, since many notions fundamental for a working argumentation system (e.g., preference criteria between conflicting arguments) are not considered. Shoham's own work at this time considered preference on models [Shoham 1988], a semantical explanation of nonmonotonicity. Nevertheless, this idea was not incorporated into the syntactic system.

3.3 Simari and Loui: Pollock and Poole Combined

Guillermo Simari and Ronald Loui [Simari 1989; Simari and Loui 1992] present a system in which a general theory of warrant (as presented by Pollock [1987]) is combined with the definition of specificity as proposed by Poole [1988]. Their main concern is to characterize, by taking into account syntactic considerations only, the conditions under which an argument structure is preferred.

The knowledge of an agent is represented by a pair \((K, \Delta)\). The set \(K\) represents indefeasible knowledge, which should be consistent (\(K \not\vdash \bot\)). \(K\) is composed of a set of necessary knowledge, \(K_N\), and a set of contingent knowledge \(K_C\). \(\Delta\) is a finite set of defeasible rules. Defeasible rules are expressed in the metalanguage. A defeasible rule of the form \(p \Rightarrow q\) is used to represent that “\(p\) is a reason for \(q\).”

Simari and Loui define a meta-meta relationship that relates formulas of \(K\) and ground instances of \(\Delta\) with a formula \(h\). This meta-meta relationship, written \(\vdash\), is called defeasible consequence, and it represents a derivation from \(K\) to \(h\) using ground instances of \(\Delta\), which can be regarded as material implications for the application of modus ponens. A subset \(T\) of ground instances of \(\Delta\)’s members is an argument for a sentence \(h\) if and only if:

1. \(K \cup T \vdash h\). \((T\) derives \(h.\))
2. \(K \cup T \vdash \bot\) \((T\) is consistent w.r.t. \(K\)).
3. \(\not\exists T' \subseteq T\) such that \(K \cup T' \vdash h\). \((T\) is minimal.)

An argument \(T\) and a formula \(h\) in the conditions defined above comprise an argument structure, written \((T, h)\). Arguments contain subarguments. The argument structure \((S, j)\) is a subargument of the argument structure \((T, h)\) if \(S \subseteq T\).

Example 3.4 (adapted from Chesñevar [1996]). Suppose we are provided with the following knowledge about cars:

- Cars that are reliable are usually recommendable.
- Cars that are reliable but no longer produced are usually not recommendable.
- European cars for which there are spare parts are usually reliable.
- Cars with cooling problems are usually not reliable.
- Cars with a rear motor usually have cooling problems.
- Cars with a rear motor usually have rear-wheel drive.
- Cars that have low consumption and rear-wheel drive are usually reliable.
- Italian cars are European cars.

and suppose we know that \(f6\) is an Italian car that has a spare parts available, has a rear motor and low consumption, and it is no longer produced. This knowledge can be modeled in the Simari–Loui framework by a set:

\[K_p = \{ \text{rear\_motor}(f6), \text{italian}(f6), \text{low\_cons}(f6), \text{no\_longer\_produced}(f6), \text{has\_spare\_parts}(f6) \}\]

of contingent knowledge, a set \(K_G = \{ \text{italian}(X) \rightarrow \text{eupean}(X) \}\) of necessary knowledge, and a set \(\Delta\) of defeasible rules, defined as follows:

\[
\Delta = \{ \\
\text{reliable}(X) \rightarrow \neg \text{recommendable}(X), \\
\text{recommendable}(X) \land \neg \text{no\_longer\_produced}(X) \rightarrow \neg \text{recommendable}(X), \\
\text{recommendable}(X) \lor \text{has\_spare\_parts}(X) \rightarrow \text{reliable}(X), \\
\text{rear\_motor}(X) \rightarrow \neg \text{system}(X), \\
\text{low\_cons}(X) \land \text{system}(X) \rightarrow \text{reliable}(X), \\
\text{rear\_motor}(X) \rightarrow \neg \text{cooling\_problems}(X), \\
\text{cooling\_problems}(X) \rightarrow \neg \text{reliable}(X) \}\]
Logical Models of Argument

Some of the different arguments that can be built from this knowledge are $A_1$ for $\text{recommendable}(f6)$, $A_2$ for $\neg\text{recommendable}(f6)$, $A_3$ for $\neg\text{reliable}(f6)$ and $A_4$ for $\text{reliable}(f6)$, where

$$A_1 = \{ \text{european}(f6) \land \text{has_spare_parts}(f6) \rightarrow \text{reliable}(f6), \text{reliable}(f6) \rightarrow \text{recommendable}(f6) \}$$

$$A_2 = \{ \text{rear_motor}(f6) \rightarrow \text{cooling_problems}(f6), \text{cooling_problems}(f6) \rightarrow \neg\text{recommendable}(f6) \}$$

$$A_3 = \{ \text{european}(f6) \land \text{has_spare_parts}(f6) \rightarrow \text{reliable}(f6), \text{reliable}(f6) \land \text{no_longer_produced}(f6) \rightarrow \neg\text{recommendable}(f6) \}$$

$$A_4 = \{ \text{rear_motor}(f6) \rightarrow \text{rear_system}(f6), \text{rear_system}(f6) \land \text{low_cons}(f6) \rightarrow \neg\text{recommendable}(f6) \}$$

An argument can be depicted as a treelike structure, as shown in Figure 2.

Arguments can be in conflict in different ways. The binary relationships of disagreement, counterargumentation, and defeat are defined as follows:

- Two arguments $\langle A_1, h_1 \rangle$ and $\langle A_2, h_2 \rangle$ disagree if and only if $K \cup \{ h_1, h_2 \} \vdash \bot$.
- Given two arguments $\langle A_1, h_1 \rangle$ and $\langle A_2, h_2 \rangle$, $\langle A_1, h_1 \rangle$ counterargues $\langle A_2, h_2 \rangle$ at literal $h$ if and only if there exists a subargument $\langle A, h \rangle$ of $\langle A_2, h_2 \rangle$ such that $\langle A_1, h_1 \rangle$ and $\langle A, h \rangle$ disagree. The subargument $\langle A, h \rangle$ of $\langle A_2, h_2 \rangle$ is called the disagreeing subargument.
- Given two arguments $\langle A_1, h_1 \rangle$ and $\langle A_2, h_2 \rangle$, $\langle A_1, h_1 \rangle$ defeats $\langle A_2, h_2 \rangle$, in the weak or strong sense, if and only if there exists a subargument $\langle A, h \rangle$ of $\langle A_2, h_2 \rangle$ such that: $\langle A_1, h_1 \rangle$ counterargues $\langle A_2, h_2 \rangle$ at the literal $h$ and (1) $\langle A_1, h_1 \rangle$ is strictly more specific\(^{14}\) than $\langle A, h \rangle$, or (2) $\langle A_1, h_1 \rangle$ is unrelated by specificity to $\langle A, h \rangle$.\(^{15}\) If $\langle A_1, h_1 \rangle$ defeats $\langle A_2, h_2 \rangle$, we also say that $\langle A_1, h_1 \rangle$ is a defeater for $\langle A_2, h_2 \rangle$.

**Example 3.5** Consider Example 3.3. Then the following relations hold\(^{16}\):

$$\langle A_3, \neg\text{reliable}(f6) \rangle$$ and $$\langle A_4, \text{reliable}(f6) \rangle$$ disagree.

$$\langle A_3, \neg\text{reliable}(f6) \rangle$$ counterargues $$\langle A_1, \text{recommendable}(f6) \rangle$$.

$$\langle A_2, \neg\text{recommendable}(f6) \rangle$$ is a proper defeater for $$\langle A_1, \text{recommendable}(f6) \rangle$$.

$$\langle A_3, \neg\text{reliable}(f6) \rangle$$ is a blocking defeater for $$\langle A_2, \neg\text{recommendable}(f6) \rangle$$.

---

\(^{14}\)Specificity imposes a partial order on arguments, preferring "more-informed" or "more-direct" arguments over other ones [Simari and Loui 1992]. Other preference criteria could also be valid.

\(^{15}\)In case 1, defeaters are called proper defeaters. In case 2, they are called blocking defeaters [Simari et al. 1994b].

\(^{16}\)Note that the relationships of disagreement, counterargumentation, and defeat are related in the following way: defeat implies counterargumentation, and counterargumentation implies disagreement.
2. An argument is active as a supporting argument at level $n + 1$ if there is no interfering argument at level $n$ such that this second argument defeats the first argument.

3. An argument is active as an interfering argument at level $n + 1$ if and only if there is no active interfering argument at level $n$ such that this second argument is strictly more specific than the first argument.

An argument $(A, h)$ that ultimately remains accepted as a supporting argument is called a justification. This inductive definition was later refined as follows.

3.3.1 The MTDR Framework: Extending Simari–Loui. Simari and Loui’s work was used as a basis for an enhanced argumentative system [Simari et al. 1994b, referred to as MTDR, in which the inductive definition of justification was recast in terms of a treelike structure, called a dialectical tree. This new conceptualization allowed the consideration of cases of fallacious argumentation. Different kinds of undesirable situations were found and settled.

Formally, a dialectical tree for $(A, h)$, denoted $T_{A,h}$, is recursively defined as follows: (1) A single node containing an argument $(A, h)$ with no defeaters is by itself a dialectical tree for $(A, h)$. (2) Suppose that $(A, h)$ is an argument with defeaters $(A_1, h_1), (A_2, h_2), \ldots, (A_n, h_n)$. We construct the dialectical tree $T_{A,h}$ by putting $(A, h)$ as the root node of it and by making this node the parent node of the roots of the dialectical trees of $(A_1, h_1), (A_2, h_2), \ldots, (A_n, h_n)$.

Any path $\lambda = [(A_0, h_0), (A_1, h_1), \ldots, (A_k, h_k)]$ in a dialectical tree $T_{A,h}$ is called argumentation line. An argumentation line $\lambda$ can be thought of as an alternate sequence of supporting and interfering arguments, starting with the supporting argument $(A_0, h_0)$. The set of supporting (interfering) arguments in $\lambda$ is denoted as $S_\lambda$ ($I_\lambda$). Supporting arguments in an argumentation line $\lambda$ belong to even levels ($0, 2, 4, \ldots$), whereas interfering arguments belong to odd levels ($1, 3, 5, \ldots$). The root of the dialectical tree is a 0-level supporting argument. The dialectical tree can be thought of as a debate between two parties, proponent and opponent. Even-level arguments (including $(A, h)$) correspond to the proponent, and they support (directly or indirectly) the hypothesis $h$, whereas odd-level arguments correspond to the opponent, who will try to defeat the proponent’s arguments.

Nodes in a dialectical tree $T_{A,h}$ can be recursively marked as undefeated nodes (U-nodes) and defeated nodes (D-nodes) as follows: (a) Leaves in $T_{A,h}$ are U-nodes; (b) Let $(B, q)$ be an inner node in $T_{A,h}$. Then $(B, q)$ will be a U-node if and only if every child node of $(B, q)$ is a D-node. $(B, q)$ will be a D-node if and only if it has at least one U-node as a child node. If the label of the root of $T_{A,h}$ (i.e., $(A, h)$) is a U-node, then $(A, h)$ is said to be a justification, or a justified argument. In that case, the hypothesis $h$ can be accepted as justified belief. Defeaters in a dialectical tree should satisfy certain constraints [Simari et al. 1994b]. Two constraints that have dialectical motivations are: (1) the tree should be cycle-free (argumentation should not be circular); (2) there should not be contradictory argumentation. The latter constraint is formally stated as demanding all odd-level (even-level) arguments in a given argumentation line to be consistent with respect to $K$.

Example 3.6 Consider the arguments in Example 3.3 and their relationships as shown in Example 3.5. Then a dialectical tree $T_{A,\text{recommendable}(f6)}$ can be built and marked as shown in Figure 3. This shows that the argument $A_1$ for recommendable$(f6)$ is not a justification.
The MTDR framework was used as a basis for further work on argumentation. Carlos I. Chesñevar [Chesñevar 1996] analyzed strategies for speeding up the inference process in MTDR by pruning the search space for arguments when determining whether an argument justifies a given literal \( h \). Chesñevar discusses a consistency-based approach for pruning the dialectical tree needed for computing justifications. This approach is based on the dialectical constraint according to which all odd-level arguments in a given argumentation line must be consistent with respect to \( K \) (the same applies for all even-level arguments). This was combined with \( \alpha - \beta \) pruning. He also considers a generalization of the notion of argument, called argument form, which allows reasoning defeasibly with nonground information, so that trees of argument forms can remain preserved as new ground facts arrive at the agent's knowledge base. The pruning criterion for dialectical trees can also be applied on trees of argument forms. Chesñevar’s recent work [Chesñevar and Simari 1998] has been focused on the formalization of defeasible argumentation using labeled deductive systems [Gabbay 1996].

Alejandro J. García defined a defeasible logic programming language (DLP), using MTDR as the inference procedure [García 1997]. DLP is an extension of (Horn-clause) logic programming, which supports programming with clauses expressed as defeasible implication. DLP captures commonsense reasoning features, allowing the representation and use of incomplete information and possibly inconsistent information. To do so, it uses an extension of MTDR which can soundly decide between contradictory goals, thereby providing a justification of any answer to a query through analysis of arguments and counterarguments. For the implementation of DLP, he developed the Justification Abstract Machine (JAM) [García 1997], that was designed as an extension of the Warren’s Abstract Machine (WAM) [Hassan 1991]. In JAM the whole process of justification is done in backward chaining and using only the information from the arguments under consideration. Thus, the complexity of answering queries is independent of the size of the program. DLP extends MTDR allowing negation as failure and presuppositions (defeasible facts). The MTDR specificity criterion was extended for comparing arguments containing these new features.

### 3.4 Vreeswijk’s Abstract Argumentation System

Gerhard Vreeswijk [1993] presents an abstract argumentation system, which results from an exhaustive analysis of the different issues arising in defeasible argumentation, such as the computational value of debate and the relation between argumentation and plausible reasoning. His system shares many features with Lin and Shoham’s approach [Lin and Shoham 1989], but incorporates some extra relationships among arguments (such as conflict, compatibility, and defeat). His approach does not deal with the problem of prescribing how argumentation should be performed, but rather with the problem of giving a formal characterization of the concepts involved in the process of argumentation.

An abstract argumentation system is a triple \((\mathcal{L}, R, \preceq)\) composed of a language \( \mathcal{L} \), a set of inference rules \( R \) and a relation \( \preceq \) between arguments. The language is composed of sentences and can be any logical language provided it contains an element representing the contradiction. The rules are not part of the language itself. Vreeswijk distinguishes between strict rules and defeasible rules. Strict rules are formulas of the form \( A_1 \ldots A_n \rightarrow A \), while defeasible rules are formulas of the form \( A_1 \ldots A_n \Rightarrow A \). The \( A_i \) and \( A \) are sentences of the underlying language. Rules can be chained together to form arguments.

**Example 3.7** Let \( \mathcal{L} = \{ p, q, r, s \} \), and let \( R = \{ p \rightarrow q; p, q \rightarrow r; q, r \Rightarrow s; r, s \rightarrow p \} \). Then \( \sigma_1 = \{ q, r \Rightarrow s \} \) and \( \sigma_2 = \{ p, q \rightarrow p \} \) are arguments.
According to Vreeswijk, arguments are proofs of varying conclusive force. He states that, besides the distinction between defeasible and strict rules, the most important difference between an argumentation system and a classical proof system is that arguments can be compared in terms of their conclusive force. The relation \( \preceq \) defines a reflexive and transitive order on arguments in terms of their conclusive force. Vreeswijk states that conclusive force is determined by other things than syntactical structure alone. He poses some constraints on the \( \preceq \) relation:

1. Infinite chains are not allowed, i.e. \( \sigma_1 < \sigma_2 < \ldots < \sigma_n < \ldots \).

2. Nonincrement of conclusive force: if \( \sigma \) is a subargument of \( \tau \), then \( \tau \preceq \sigma \) for every \( \sigma \) and \( \tau \).

3. Strict propagation of conclusive force: if \( \sigma_1 \ldots \sigma_n \rightarrow \phi \), then \( \sigma_i \preceq \phi \) for some \( 1 \leq i \leq n \).

The ultimate aim of the process of argumentation is to determine which arguments are in force. In order to characterize the concept of argument in force, and therefore the concept of warranted conclusion, he defines a theory of warrant. A theory of warrant declares which arguments are active, and which conclusions are warranted, with respect to a set of ground information called the base set. In order to avoid conflicts between strict arguments that cannot be defeated, a base set should be a finite and compatible subset of \( \mathcal{L} \). A subset \( P \) of \( \mathcal{L} \) is incompatible if there exists a strict argument with conclusion \( \bot \).

**Example 3.8** [Vreeswijk 1993]. Let \( \mathcal{L} = \{ p, q, r, s \} \cup \{ \bot \} \), and let \( R = \{ p \Rightarrow q ; p, q \Rightarrow s ; r, s \Rightarrow \bot \} \). Then all subsets of \( \{ p, q, s \} \), \( \{ p, r \} \), and \( \{ q, r \} \) are compatible, while all supersets of \( \{ p, q, s \} \) and \( \{ r, s \} \) are not compatible.

The most elementary notion of warrant is the notion of enablement. Given a fixed set of arguments \( \Sigma \), an argument \( \sigma \) is enabled by \( \Sigma \) if and only if it is not defeated by an argument (or group of arguments) in \( \Sigma \). As a special case, any element of the base set or any strict consequence of a set of enabled arguments is enabled. In other words, an argument \( \sigma \) is enabled by \( \Sigma \) on the basis of a base set \( P \) if and only if \( \Sigma \) leaves enough room for “developing” \( \sigma \) from \( P \).

**Example 3.9** Consider the language \( \mathcal{L} = \{ p, q, r, s \} \cup \{ \bot \} \), with rules \( R = \{ p \Rightarrow q, q \Rightarrow r, p \Rightarrow s \} \cup \{ q, s \Rightarrow \bot, p \Rightarrow t \} \), and with an order on arguments \( \preceq \) such that \( p \Rightarrow q \sim p \Rightarrow s \).

Let \( P = \{ p \} \) be a base set. Then the arguments \( \sigma_1 = \{ p \} \) and \( \sigma_2 = \{ p \Rightarrow t \} \) are enabled by \( R \) on the basis of \( P \). The argument \( \sigma_3 = \{ p \Rightarrow q \} \) is not enabled, since it is defeated by \( \{ p \Rightarrow s \} \). The argument \( \sigma_4 = \{ p \Rightarrow q, q \Rightarrow r \} \) is not enabled, since it was developed from a subargument \( \{ p \Rightarrow q \} \) that was not enabled.

Vreeswijk then presents a theory of inductive warrant, based on the idea that defeaters can in their turn be defeated. This idea leads to a characterization similar to the one presented by Pollock [1987], in which arguments can be thought to be active at different levels. Vreeswijk defines levels as follows:

- If \( P \) is a base set, an argument \( \sigma \) is in at level 1 on basis of \( P \) if \( \sigma \) is based on \( P \).
- An argument \( \sigma \) is in at level \( n \) on basis of \( P \), written \( \sigma \mid P \) = 1, if \( \sigma \) is based on \( P \), if all proper subarguments of \( \sigma \) are in at level \( n \) and if \( \sigma \) has no defeaters on basis of \( P \) that are in at level \( n - 1 \).
- An argument \( \sigma \) is out at level \( n \) on basis of \( P \), written \( \sigma \mid P \) = 0, if it is not in at level \( n \) on basis of \( P \).
- The development of \( \sigma \) on basis of \( P \), written \( \{ \sigma \mid P : n \} \) is the sequence \( \{ \sigma \mid P : n \} \).

Vreeswijk links the notions of enablement and inductive warrant, and defines three classes of arguments:

- an argument \( \sigma \) is ultimately undefeated on basis of \( P \) if \( \lim_n \sigma \mid P \) = 1.
- an argument \( \sigma \) is ultimately defeated on basis of \( P \) if \( \lim_n \sigma \mid P \) = 0.
• an argument \( \sigma \) is provisionally defeated on basis of \( P \) if \( \lim_n | \sigma_n \) does not exist.

**Example 3.10** [Vreeswijk 1993]. Consider the language \( \mathcal{L} = \{ p, q, r \} \cup \{ \perp \} \), with rules \( R = \{ p \rightarrow q, p \rightarrow r \} \cup \{ q, r \rightarrow \perp \} \), and with an order on arguments \( \leq \) such that \( p \rightarrow q \prec p \rightarrow r \) and \( p \rightarrow r \prec p \rightarrow q \).

Let \( P = \{ p \} \) be a base set, and let \( \sigma_1 = \{ p \rightarrow q \} \) and \( \sigma_2 = \{ p \rightarrow r \} \). Since both of them are based on \( P \), it holds that they are in at level 1, that is, \( | \sigma_1 |_1 = 1 \) and \( | \sigma_2 |_1 = 1 \). To see whether \( \sigma_1 \) and \( \sigma_2 \) are at level 2, we search for defeaters for them at level 1. Since \( \sigma_1 \) defeats \( \sigma_2 \), and \( \sigma_2 \) defeats \( \sigma_1 \), it follows that both are out at level 2, (\( | \sigma_1 |_2 = 0 \) and \( | \sigma_2 |_2 = 0 \)). Because of this, it follows that both arguments will be in at level 3 (\( | \sigma_1 |_3 = 1 \) and \( | \sigma_2 |_3 = 1 \)) and once again out at level 4 (\( | \sigma_1 |_4 = 0 \) and \( | \sigma_2 |_4 = 0 \)). This yields \( | \sigma_1 |_p = \langle 1, 0, 1, 0, \ldots \rangle \), and the same applies for \( \sigma_2 \). It follows that both \( \sigma_1 \) and \( \sigma_2 \) are provisionally defeated.

Finally, Vreeswijk introduces the notion of warrant, in which no inductive definition is needed, by considering a set of arguments to be in force at different levels. To obtain the arguments that are in force at level \( n \) the enable operator should be applied \( n \) times to a given base set. A conclusion of an argument that is in force at level \( n \) is warranted at level \( n \). The warrant relation can yield more than one fix-point. Therefore multiple extensions can be obtained from a given base set. Vreeswijk presents a non constructive, fixed-point definition for extension, and also shows how the notions of enablement, warrant and inductive warrant are linked to each other.

Vreeswijk also formalizes the notion of defeasible debate as an alternative to determining whether a specific thesis \( \phi \) is tenable relative to a fixed base set \( P \), assuming an abstract argumentation system \( A \) in the background. This debate is carried out by a proponent \( P \) and opponent \( O \), who may perform three kinds of moves: they may state one or more theses, along with supporting arguments, pose a question, or give an answer. The party stating the theses is the defender of them; the other party is the attacker. Both \( P \) and \( O \) may take these positions. A debate turns out to be a (possibly infinite) sequence of moves such that every thesis (or list of theses) is followed by a question, every question is followed by an answer, and every answer is, if possible, followed by a thesis (or list of theses). Vreeswijk shows how this formalization of defeasible debate fits in the theory of level-\( n \) arguments. If \( P \) is a base set, and \( \sigma \) is an argument based on \( P \), then it holds that

• The argument \( \sigma \) is ultimately defeated on the basis of \( P \) if and only if \( \sigma \) can be defended successfully in finite depth in every debate on the basis of \( P \).

• The argument \( \sigma \) is ultimately defeated on the basis of \( P \) if and only if \( \sigma \) can be attacked successfully in finite depth in every debate on the basis of \( P \).

In defeasible debates, there is always a winner and a loser. The winner is the party that has made the last move. Vreeswijk however distinguishes two ways of winning a debate: winning on substantial grounds (the winning party gave an answer to which the other party was unable to respond), and winning on procedural grounds (the other party was able to respond, but did not do so, either because it was not permitted by procedure (e.g., lack of time), or because the loser decided to quit.

These considerations lead Vreeswijk to analyze the role of limited resources in the process of argumentation, and the meaning of provisional conclusions resulting from partially completed debates. The problem arises from arguments that are still open (i.e., it is unclear whether their conclusions must be accepted or rejected). Within his formal framework, he shows that the intermediate results of a running debate approximate the final outcome, and that the quality of these approximations is entirely determined by the criterion according to which arguments are accepted or rejected.

Vreeswijk implemented IACAS, a program to do interactive argumentation on a computer [Vreeswijk 1994]. The program
is implemented in Lisp, and it is intended to demonstrate Vreeswijk's theory of argumentation. Later, Vreeswijk extended some of his results into novel areas, such as multiagent systems [Vreeswijk 1995b] and meta-games for changing the rules of argument games [Vreeswijk 1995a; 1999].

3.5 Brewka's Approach: Rescher and Reiter Combined

Gerhard Brewka [Brewka 1994b] presents a formal reconstruction of Rescher's theory of disputation [Rescher 1977]. His approach incorporates some elements of Reiter's Default Logic [Reiter 1980], where Rescher's provisoed assertions are seen as Reiter's defaults. The notion of extension is also borrowed from Reiter, and it is modified in order to define SDL extensions. SDL extensions are default extensions that deal with specificity. A disputation about a formula \( P \) takes the form of a process in which the goal of the proponent is to construct a default theory that implies \( P \) and is accepted by the opponent, while the aim of the opponent is to question the proponent's claims or to add new evidence against them.

Next we will briefly review Reiter's Default Logic (DL) and a variant of default logic developed by Brewka [1994a; 1994b], which will be used as a basis for reconstructing Rescher's theory. In DL, default theories consist of a set of facts \( W \) and a set of defaults \( D \). Each default is of the form \( A : B_1, \ldots, B_n/C \). A default theory generates extensions, which are defined as fixed points of an operator \( \gamma \). This operator maps an arbitrary set of formulas \( S \) to the smallest deductively closed set \( S' \) that contains \( W \) and satisfies the following condition: if \( A : B_1, \ldots, B_n/C \in D, A \in S' \), and \( \forall i, i = 1 \ldots n, \neg B_i \notin S \), then \( C \in S' \). In DL extensions represent sets of acceptable beliefs a reasoner might adopt. Skeptical inference can be characterized by considering a formula to be provable if it is contained in all extensions of \( (D, W) \).

Default logic does not prefer more specific defaults over more general ones. However, Rescher's use of provisoed assertions involves specificity as preference criterion. This motivates Brewka to extend DL, developing PDL (a prioritized version of DL) and SDL (DL with specificity). Both PDL and SDL are defined for normal defaults only.\footnote{Normal defaults have the form \( A : B/B \).} A normal default \( A : B \rightarrow B \) corresponds to Rescher's provisoed assertion \( B \rightarrow A \). To avoid confusion, Brewka writes \( A \rightarrow B \) to denote the normal default \( A : B \rightarrow B \). Next we will introduce some definitions that characterize both PDL and SDL.

Given a set of formulas \( E \), and a default \( \delta = a \rightarrow b, \delta \text{ is active in } E \text{ iff (1) } a \in E, (2) b \notin E, \text{ and (3) } \neg b \notin E \).

A (prioritized) default theory is defined as a triple \( (D, W, <) \). Given a strict total order \( > \) containing \( < \), the set \( E \) is a (PDL) extension of \( \Delta \) generated by \( > \) iff \( E = \bigcup E_i \), where \( E_0 := \text{Theorem}(W) \), and

\[
E_{i+1} = \begin{cases} 
E_i & \text{if no default is active in } E_i \\
\text{Theorem}(E_i \cup \{b\}) & \text{otherwise, where } b \text{ is the consequent of the}
\end{cases}
\]

\( \Rightarrow \) minimal default \( \rightarrow \) that is active in \( E_i \)

If \( \Delta = (D, W, <) \) is a (prioritized) default theory, we will say that \( E \) is a (PDL) extension of \( \Delta \) iff there is a strict total order containing \( < \) that generates \( E \).

In order to define the priority ordering representing specificity, Brewka distinguishes in \( W \) between background knowledge \( T \) and contingent facts \( C \) as usual in conditional approaches. A default theory \( (D, W, <) \) is therefore split to a theory \( (D, T, C) \). Only the sets \( T \) and \( D \) will be used to determine specificity. For a given set of defaults \( D \), Brewka introduces the notation \( P \vdash_D q \) to express that \( q \)
is contained in the smallest deductively closed set of wffs that contains \( P \) and is closed under \( D \).

If \( \Delta = (D, T, C) \) is a default theory, the set \( D' \subseteq D \) is p-conflicting (in \( \Delta \)) if for some \( \delta = \text{pre} \rightarrow \text{cons} \in D' \) it holds that

\[
T \cup \{\text{pre}\} \not\vdash_{D} \text{false}
\]

Given a default theory \( \Delta = (D, T, C) \), \( <_{\Delta} \) is the specificity ordering associated with \( \Delta \) if \( <_{\Delta} \) is the smallest strict partial ordering on \( D \) satisfying the condition that

\[
\delta_1 = \text{pre}_1 \rightarrow \text{cons}_1 <_{\Delta} \delta_2 = \text{pre}_2 \rightarrow \text{cons}_2,
\]

whenever

1. \( \delta_1, \delta_2 \) are contained in a minimal set \( D' \) of defaults p-conflicting in \( \Delta \), and
2. \( T \cup \{\text{pre}_1\} \not\vdash_{D'} \text{false} \), but \( T \cup \{\text{pre}_2\} \not\vdash_{D'} \text{false} \).

Finally, we will say that \( E \) is an SDL extension of \( \Delta = (D, T, C) \) if the specificity ordering \( <_{\Delta} \) of \( \Delta \) exists and \( E \) is a prioritized extension of \( (D, T \cup C, <_{\Delta}) \).

Following Rescher, Brewka takes defaults as shared knowledge between proponent and opponent. Some particular sets must be distinguished:

- \( D = \text{defaults} \) supported by both proponent and opponent.
- \( T = \text{background knowledge supported by both proponent and opponent.} \)
- \( C = \text{contingent facts shared by both proponent and opponent.} \)
- \( T_{\text{pro}} \) and \( C_{\text{pro}} \) (\( T_{\text{opp}} \) and \( C_{\text{opp}} \)): background knowledge and contingent facts supported by the proponent (opponent).

Brewka denotes as \( \Delta_{\text{pro}} \) (\( \Delta_{\text{opp}} \)) the current default theory supported by the proponent (opponent). The proponent’s theory will be defined as \( (D, T \cup T_{\text{pro}}, C \cup C_{\text{pro}}) \) (analogously for the opponent). A state in a disputation can be identified with a tuple consisting of these sets. Each player holds a default theory composed of the set of defaults, the shared knowledge, and the unshared propositions believed by the player. During the disputational process three elementary moves can take place: additions, concessions, and removals.

An addition consists of adding a formula to the current player’s background or contingent knowledge or adding a default to the set of stated defaults. A concession consists of moving a formula from the other player’s background or contingent knowledge to the set of background or contingent shared knowledge, respectively. A removal consists of discarding formulas from the current player’s background or contingent knowledge. Formally:

- \( \text{add}_p(i) \): if \( i \) is a formula, then \( g \) should be either \( T \) or \( C \). In the first case, \( i \) is added to the current player’s background knowledge, in the second case to its contingent facts. If \( i \) is a default, then \( g \) must be \( D \) and \( i \) is added to \( D \).
- \( \text{concede}_p(i) \): moves formula \( i \) from the other player’s background knowledge (if \( g = T \)) or from the other player’s contingent facts to \( C \) (if \( g = C \)).
- \( \text{remove}_p(i) \): removes formula \( i \) from the current player’s background knowledge (if \( g = T \)) or from the current player’s contingent facts (if \( g = C \)).

A move is an arbitrary finite sequence of elementary moves. For a move to be consistent, the current player’s theory must have at least one consistent extension. For a move to be legal it has to be consistent and the new state has to be different from any previous state. The last condition ensures that the argumentation process will be progressive.

A debate about a formula \( P \) is won by the proponent if after a move of the opponent, \( P \) belongs to all the SDL extensions of the opponent’s theory. The opponent wins the debate about a formula \( P \) if after a move of the proponent, \( P \) is not in all the SDL extensions of the proponent’s theory.

According to Brewka’s approach, a debate could be carried out forever (in spite of forbidding repetitions). Hence he considers it reasonable to assume an additional, nonlogical termination condition (e.g., time limit) for the game.

Example 3.11 [Brewka 1994b]. Consider Example 3.1. This is a disputation
about the flying ability of birds in terms of Brewka’s approach:\(^{19}\):

\[
\begin{array}{c|c}
\text{Proponent} & \text{Opponent} \\
\hline
\text{(1)} & add_C(\neg \text{Flies}) \\
\text{(2)} & add_D(\text{Bird} \rightarrow \text{Flies}) \\
\text{(3)} & \text{concede}_C(\text{Bird}) \\
& add_D(\text{Penguin} \rightarrow \neg \text{Flies}) \\
& add_C(\text{Penguin}) \\
& add_T(\text{Penguin} \supset \text{Bird}) \\
\text{(4)} & \text{concede}_T(\text{Penguin} \supset \text{Bird}) \\
& add_C(\neg \text{Penguin}) \\
\text{(5)} & add_D(\text{Bird} \land \text{Swims} \rightarrow \text{Penguin}) \\
& add_C(\text{Swims}) \\
\text{(6)} & \text{concede}_C(\text{Swims}) \\
& retract_C(\neg \text{Penguin})
\end{array}
\]

After the last move of the opponent, the following situation results:

\[
D = \{ \text{Bird} \rightarrow \text{Flies}, \text{Penguin} \rightarrow \neg \text{Flies}, \text{Bird} \land \text{Swims} \rightarrow \text{Penguin} \} \\
C = \{ \text{Bird}, \text{Swims} \} \\
T = \{ \text{Penguin} \supset \text{Bird} \}
\]

In this stage of the debate, both \( T_{\text{opp}} \) and \( C_{\text{opp}} \) are empty, and the proponent’s thesis is a skeptical conclusion of \( \Delta_{\text{opp}} \). Therefore this is a winning situation for the proponent; the disputation stops without the need to ask for a third-party determiner.

Recently Brewka has developed a formal model of argumentation that captures both the logical and the procedural aspects of argumentation processes [Brewka 1999]. His model is based on situation calculus [McCarthy 1963; Levesque et al. 1998]. In that model he shows how metalevel argumentation can be modeled through dynamic argument systems, in which situation calculus plays a central role in the formalization of structured debates.

3.6 Dung’s Argumentation Theory: Extending Logic Programming

Phan Minh Dung [Dung 1993a; 1993b] develops an argumentation theory whose central notion is the acceptability of arguments. He shows that most of the major approaches to nonmonotonic reasoning in AI and logic programming turn out to be special forms of his theory of argumentation. Dung also shows how his theory can be used to capture the logical structure of many practical problems, such as the theory of \( n \)-person games and the stable marriage problem.

Dung [1993b] seeks to provide a framework for the semantics of extended logic programming with explicit negation. An extended logic program \( P \) is composed of clauses of the form:

\[
L_0 \leftarrow L_1, \ldots, L_m, \text{not-}L_{m+1}, \ldots, \text{not-}L_{m+n},
\]

where \( L_i \)'s are objective literals. An objective literal can take the form \( A \) or \( \neg A \), where \( A \) is an atom. Literals of the form not-\( L_i \) are called subjective literals and are the assumptions on which the proof is based. Since the acceptance of a proof depends on the acceptance of the assumptions on which it is based, arguing

\(^{19}\)In this example we use Brewka’s notation. The symbols \( \supset \) and \( \rightarrow \) stand for material and default implication, respectively.
for a conclusion means arguing for the assumptions on which some proof of the conclusion is based. Therefore, arguments can be viewed as sets of assumptions, or equivalently, as sets of ground subjective literals. An argument \( A \) supports an objective literal \( L \) if there exists a proof of \( L \) based on assumptions contained in \( A \). An argument is self-defeating if it supports both \( L \) and \( \neg L \), where \( L \) is an objective literal.

Dung distinguishes two ways of attacking an argument:

- An argument \( A \) attacks an argument \( A' \) via reductio ad absurdum (RAA-attack) if \( A \cup A' \) is self-defeating.
- An argument \( A \) attacks an argument \( A' \) via ground attack (\( g \)-attack) if there is an assumption not-\( L \) in \( A' \) such that \( L \) is supported by \( A \).

**Example 3.12** Consider the following extended logic program:

\[
\begin{align*}
\text{innocent}(X) &\leftarrow \neg \text{guilty}(X) \\
\text{guilty}(X) &\leftarrow \text{confess_guilt}(X) \\
\neg \text{innocent}(X) &\leftarrow \text{confess_guilt}(X), \\
&\quad \text{not lies}(X)
\end{align*}
\]

Consider the sets of assumptions \( A_0 = \emptyset \), \( A_1 = \{ \text{not-guilty}(\text{john}) \} \), \( A_2 = \{ \text{not-lies}(\text{peter}) \} \), and \( A_3 = \{ \text{not-guilty}(\text{peter}) \} \). Then the argument \( A_0 \) supports \( \text{guilty}(\text{john}) \), the argument \( A_1 \) supports \( \text{innocent}(\text{john}) \), the argument \( A_2 \) supports \( \neg \text{innocent}(\text{peter}) \).

Note that the argument \( A_2 \) for \( \neg \text{innocent}(\text{peter}) \) RAA-attacks the argument \( A_3 \) for \( \text{innocent}(\text{peter}) \), since \( A_2 \cup A_3 \) is self-defeating, supporting both \( \text{innocent}(\text{peter}) \) and \( \neg \text{innocent}(\text{peter}) \). The argument \( A_0 \) for \( \text{guilty}(\text{john}) \) is a ground attack for the argument \( A_1 \) for \( \text{innocent}(\text{john}) \).

Given an extended logic program \( P \), the semantics of \( P \) is defined by the semantics of the argumentation framework:

\[
AF(P) = \langle AR_P, \text{attacks}, g\text{-attacks} \rangle
\]
such that the following conditions are satisfied:

a) \( AR_P \) denotes the set of all sound arguments w.r.t. \( P \).

b) \( \text{attacks} = \text{RAA-attacks} \cup \text{g-attacks} \)

c) \( \text{RAA-attacks}(A', A) \) if and only if \( A' \) represents a RAA-attack against \( A \).

d) \( \text{g-attacks}(A', A) \) if and only if \( A' \) represents a ground attack against \( A \).

An argumentation framework is a triple

\[
AR = \langle A, \text{attacks}, g\text{-attacks} \rangle
\]
where \( A \) is a set of arguments, and attacks and g-attacks are binary relations such that g-attacks \( \subseteq \) attacks.

When a set of arguments \( S \) is such that no two elements of \( S \) attack each other, then \( S \) is said to be conflict-free. An argument \( A \) is acceptable with respect to a set of arguments \( S \) if for any argument \( B \) that attacks \( A \), \( B \) is g-attacked by some argument \( A' \in S \). A conflict-free set of arguments \( S \) is admissible if each argument in \( S \) is acceptable with respect to \( S \).

Dung then defines the credulous semantics of an argumentation framework using the notion of preferred extension. A preferred (credulous) extension of an argumentation framework \( AF \) is a maximal (w.r.t. set inclusion) admissible set of arguments. Dung shows that preferred extension semantics is always defined for argumentation frameworks.

The notion of Dung’s stable extension is the counterpart of the answer-set semantics of extended logic programs [Gelfond and Lifschitz 1990]. A conflict-free set of arguments \( S \) is a stable extension if \( S \) g-attacks each argument that does not belong to \( S \). Every stable extension is a preferred extension, but the converse does not hold in general. A set \( S \) is an answer-set of an extended logic program \( P \) if and only if there is a stable extension \( E \) of \( AF(P) \) such that

\[
S = \{ L | L \text{ is supported by an argument from } E \}
\]

Dung introduces the ground (skeptical) semantics in terms of a fixpoint theory.
The characteristic function of an argumentation framework $AF$, denoted by $F_{AF}$, such that $F_{AF}: 2^{AR} \to 2^{AR}$ is defined as $F_{AF}(S) = \{A | A$ is acceptable w.r.t. $S\}$.

Dung shows that $F_{AF}$ is monotonic with respect to set inclusion, and that a conflict-free set $S$ of arguments is admissible if and only if $S \subseteq F_{AF}(S)$. The grounded extension of an argumentation framework $AF$ is the least fixed point of $F_{AF}$. Grounded extensions can be used as the basis to characterize the semantics of normal logic programming.

The notion of complete extension provides the link between credulous and skeptical semantics, that is, between preferred extensions and grounded extensions. An admissible set of arguments $S$ is called a complete extension if and only if each argument which is acceptable with respect to $S$ belongs to $S$. Dung presents a theorem showing that each preferred extension is a least complete extension, and the grounded extension is a least (with respect to set inclusion) complete extension.

Later, Dung [1995b] proposes an even more abstract argumentation framework in which only one binary relation representing the attack relationship between arguments is considered. The new theory of argumentation shares most features with the approach just described. However, the author goes beyond the previous results by showing that his theory of argumentation provides a unified foundation for different well-known approaches to nonmonotonic reasoning. According to Dung [1995b]:

This result is not as surprising as it seems since all forms of reasoning with incomplete information rest on the simple intuitive idea that a defeasible statement can be believed only in the absence of any evidence to the contrary which is very much like the principle of argumentation.

Example 3.13 [Dung 1995b]. To illustrate this abstract argumentation framework Dung proposes the following disputation between two persons $I$ and $A$:

$I$: My government cannot negotiate with your government because your government doesn’t even recognize my government.

$A$: Your government doesn't recognize my government which is a terrorist government.

$I$: But your government is a terrorist government.

Assume $i_1$ and $i_2$ denote the first and second argument of $I$, respectively, and $a$ denotes the argument of $A$. The above disputation can be represented by an argumentation framework $AF = \langle AR, attacks \rangle$ as follows:

$$AR = \{i_1, i_2, a\}$$

$$attacks = \{(i_1, a), (a, i_1), (i_2, a)\}$$

It is easy to verify that $AF$ has one preferred extension $E = \{i_1, i_2\}$. At the same time we can see that the grounded extension of $AF$ coincides with its preferred extension since $F_{AF}(\emptyset) = \{i_2\}$, $F_{AF}(\{i_1\}) = \{i_1, i_2\}$, and $F_{AF}(\{i_1, i_2\}) = (\emptyset)$.

Dung presents another example to illustrate the fact that it is possible to obtain two preferred extensions.

Example 3.14 The Nixon diamond example can be specified by the arguments $A$ and $B$, where $A$ establishes that “Nixon is nonpacifist since he is a republican,” and $B$ establishes that “Nixon is a pacifist since he is a quaker.” This example can be represented by an argumentation framework $AF = \langle AR, attacks \rangle$ with $AR = \{A, B\}$ and $attacks = \{(A, B), (B, A)\}$. This argumentation framework has a preferred extension in which Nixon is nonpacifist and another preferred extension in which Nixon is a pacifist.

Finally, it should be noted that Dung’s work made an important contribution for the development of a more evolved, refined formalism, called abstract argumentation framework, which will be discussed in Section 3.9.

3.7 Prakken and Sartor’s Framework for Legal Argumentation

Henry Prakken and Giovanni Sartor [Prakken and Sartor 1996b] study different logical notions of legal arguments by
presenting a system of defeasible argumentation or argumentation framework. The underlying language for the system contains strong literals and weak literals. Strong literals are atomic first-order formulas, which can be preceded by classical negation. Weak literals take the form \( \sim L \), where \( L \) is a strong literal. Informally, \( \sim L \) reads as “there is no evidence that \( L \) is the case,” while \( \neg L \) says “\( L \) is definitely not the case.” Weak negation is used to represent assumptions.

Rules have the form

\[
 r : L_0 \land \cdots \land L_j \land \sim L_k \land \cdots \land \sim L_m \Rightarrow L_n
\]

where \( r \) is the name of the rule, and each \( L_i \) is a strong literal. In the metalanguage, the complement of a literal \( L \) will be denoted \( \overline{L} \).

Prakken and Sartor present two versions of their framework: the first version considers fixed priorities, whereas the second one handles defeasible priorities.

### 3.7.1 Framework for Fixed Priorities

The input information of Prakken and Sartor’s framework is called an ordered theory, and contains rules divided into two categories: a set \( D \) of defeasible rules, and a set \( S \) of strict rules. Strict rules represent information intended to be beyond debate, and are composed only by strong literals. Defeasible rules represent information subject to debate, and can contain assumptions (weak literals) in their antecedents.

In the fixed-priority framework, the system takes as input an ordered theory \( (S, D, <) \), where \( < \) is a strict partial order on \( D \). An argument is a finite sequence \( A = [r_0, \ldots, r_n] \) of ground instances of rules, such that for every \( i (0 \leq i \leq n) \), for every strong literal \( L_j \) in the antecedent of \( r_i \), there is a \( k < i \) such that \( L_j \) is the consequent of \( r_k \), and no two distinct rules in the sequence have the same consequent. A literal \( L \) is a conclusion of \( A \) if and only if \( L \) is the consequent of some rule in \( A \). For any ordered theory \( \Gamma \), the set of all arguments based on \( \Gamma \) will be denoted \( \text{Args}_\Gamma \).

Conflict among arguments is defined as follows. An argument \( A_1 \) attacks \( A_2 \) if and only if there are sequences \( S_1, S_2 \) of strict rules such that: \( A_1 + S_1 \) is an argument with a conclusion \( L \), and (a) either \( A_1 + S_1 \) is an argument with a conclusion \( L \), or (b) \( A_2 \) is an argument with an assumption \( L \). An argument is coherent if it does not attack itself. A set of arguments \( \text{Args} \), it is said to be conflict-free if there are no arguments \( A, B \in \text{Args} \) such that \( A \) attacks \( B \).

Two cases of attack are distinguished, rebutting attack and undercutting attack. An argument rebuts another argument if they yield complementary conclusions and a preference for the first argument over the second one can be induced from the priorities defined on the rules that comprise the arguments. An argument undercut another argument if the first argument has a conclusion which is the strong negation of an assumption of the second argument.

The notion of defeat combines the two kinds of conflicts between arguments discussed above. Given the arguments \( A_1 \) and \( A_2, A_1 \) defeats \( A_2 \) if and only if one of the following three conditions holds:

1. \( A_1 \) is empty and \( A_2 \) is incoherent, or
2. \( A_1 \) undercut \( A_2 \), or
3. \( A_1 \) rebuts \( A_2 \) and \( A_2 \) does not undercut \( A_1 \).

We will say that \( A_1 \) strictly defeats \( A_2 \) if and only if \( A_1 \) defeats \( A_2 \) and \( A_2 \) does not defeat \( A_1 \). It must be observed that the notion of defeat is weak, in the sense that to rebut an argument we do not demand the preference relationship \( > \) between rules to hold, but \( < \) instead.

An argument \( A \) is acceptable with respect to a set of arguments \( \text{Args} \) if and only if every argument that defeats \( A \) is strictly defeated by another argument in \( \text{Args} \). The definition of acceptable arguments leads us to the notion of justified arguments (i.e., those arguments that turn out to be acceptable when all possible

---

\( \text{If } A \text{ is an argument, and } T \text{ a sequence of rules, then } A + T \text{ denotes the concatenation of } A \text{ and } T. \)
pairwise comparisons between conflicting arguments are taken into account).

The set $\text{JustArg}_r$ of all justified arguments for an ordered theory $\Gamma$ is defined in terms of a fixpoint operator, called the characteristic function of the ordered theory. If $\Gamma$ is an ordered theory, and $S \subseteq \text{Args}_r$, then the characteristic function of $\Gamma$ is defined as:

- $F_\Gamma : \text{Pow(Args}_r) \rightarrow \text{Pow(Args}_r)$
- $F_\Gamma(S) = \{ A \in \text{Args}_r \mid A$ is acceptable w.r.t. $S\}$

The $F_\Gamma$ operator can be shown to be monotonic, hence it is guaranteed to have a least fixpoint. If $A$ is an argument, it is said that on the basis of $\Gamma$:

- $A$ is justified if and only if $A$ is in the least fixpoint of $F_\Gamma$ (denoted $\text{JustArg}_r$).
- $A$ is overruled if and only if $A$ is attacked by a justified argument.
- $A$ is defensible if and only if $A$ is neither justified nor overruled.

The authors show how to capture the definition of $\text{JustArg}_r$, in a constructive way, by defining the following sequence of subsets of $\text{Args}_r$:

- $F^1 = F_\Gamma(\emptyset)$
- $F^{i+1} = F^i \cup F_\Gamma(F^i)$

It can be shown that for $\Gamma$ finite, it holds that $\bigcup_{i=0}^{\infty}(F^i) = \text{JustArg}_r$.

Example 3.15 (adapted from Prakken and Sartor [1996]). Consider the following rules in Prakken–Sartor formalism:

- $r_0: \Rightarrow X$ does not respect the law
- $r_1: X$ does not respect the law
  $\Rightarrow X$ should go to jail
- $r_2: \sim (X$ should go to jail$)$
  $\Rightarrow X$ is a good citizen
- $r_3: \Rightarrow \neg (X$ does not respect the law$)$

where $r_0 < r_3$. We can build the argument $A_1 = [r_0, r_1]$ for concluding that $X$ should go to jail, and the argument $A_2 = [r_2]$ for concluding that $X$ is a good citizen. The argument $A_1$ defeats $A_2$ by undercutting it. Furthermore, the argument $A_3 = [r_3]$ defeats $A_1$, by rebutting the proper subargument $A_0 = [r_0]$.

The set of justified arguments can be built as follows. $A_3$ is the only argument that is not defeated by any other argument, it is acceptable with respect to the empty set. Hence $F^1 = \{ [], A_3 \}$. Note that neither $A_1$ nor $A_0$ is acceptable with respect to $F^1$, since they are defeated by $A_3$, which is not strictly defeated by any argument in $F^1$. The argument $A_2$ is defeated by its only counterargument $A_1$, which is in turn strictly defeated by the argument $A_3$ in $F^1$. As a result, $A_3$ reinstates $A_2$, and $F^2 = \{ [], A_3, A_2 \}$.

Repeating this process does not add any new relevant argument, so that $F^k = F^2$, $k > 2$. Hence the set of justified arguments is $\{ A_2, A_3 \}$. Note that both $A_0$ and $A_1$ are overruled.

Prakken and Sartor [Prakken 1998] also introduce a dialectical proof theory for an argumentation framework fitting the abstract format developed by Dung, Kowalski, and others [Bondarenko et al. 1993; Dung 1993b]. A proof of a formula takes the form of a dialogue tree, where each branch of the tree is a dialogue and the root of the tree is an argument for the formula. The idea is that every move in a dialogue consists of an argument based on the input theory, where each stated argument attacks the last move of the opponent in a way that meets the player’s burden of proof. The required force of each move depends on who states it, and it is motivated by the definition of acceptability (e.g., the proponent wants a conclusion to be justified, hence his moves have to be strictly defeating; the opponent wants to prevent a conclusion from being justified, hence his moves may be just defeating). An argument $A$ is provably justified if and only if there is a dialogue tree with $A$ as its root, won by the proponent (a player wins a dialogue tree if and only if it wins all branches of the tree). Prakken and Sartor also show that all provably justified arguments are justified, and that if an argument is provably justified, then all its subarguments are provably justified.
Example 3.16 [Prakken and Sartor 1996b]. Assume we have the following input theory.

\[ r_1: \sim \neg e \text{ is admissible evidence } \Rightarrow e \text{ proves guilt } \]
\[ r_2: \Rightarrow f \text{ forged evidence } e \]
\[ r_3: f \text{ forged evidence } e \Rightarrow \neg e \text{ is admissible evidence } \]
\[ r_4: f \text{ is police officer in LA } \land \sim \neg f \text{ forged evidence } e \]
\[ r_5: f \text{ is police officer in LA } \Rightarrow f \text{ is police officer, } f \in R \]
\[ r_6: f \text{ is police officer in LA } \Rightarrow f \text{ is police officer, } r_5 \in R \]
\[ r_7: f \text{ is police officer in LA } \Rightarrow f \text{ is dishonest } \]
\[ r_8: x \text{ is police officer } \land x \text{ has received a medal of honor } \Rightarrow \neg f \text{ is dishonest } \]

which is a strictly defeating rebuttal for \([f_5, r_7]\).

After that the opponent has run out of moves, since no argument on the basis of the given input theory can defeat the proponent’s last argument. Hence the proponent succeeded in defending its argument against every possible way of attack, so that \([r_1]\) can be considered a provably justified argument.

3.7.2 Framework for Defeasible Priorities. Prakken and Sartor declare that for a theory of defeasible argumentation to be realistic it should also formalize arguments about priorities. This gives rise to their second argumentation framework, in which priorities are not fixed but defeasible. In the new approach the priority relation among rules is established by a predicate symbol \(\prec\) in the object language. This makes the third component of an ordered theory redundant, so an input theory in the new framework is just a pair \((S, D)\).

A set of fixed strict rules is added to every input theory so that the ordering on rules is a strict partial order and a formal connection between object level and metalevel is defined in such a way that priorities derived at the object level can be directly lifted to the metalevel. It is necessary that the ordering component \(\prec\) of a theory \(T\) (as defined in the framework for fixed priorities) be determined by the set of all priority arguments justified on the basis of \(T\), that is, that \((r, r') \in \prec\) if and only if there is a justified argument for \(r \prec r'\).
The idea is to construct the set of justified arguments as before, in a step-by-step fashion. In order to do so, the following notation is introduced to capture what a certain set of arguments says about the priorities. If \( \text{Args} \) is a set of arguments, then \( <_{\text{Args}} = \{ r < r' \mid r < r' \} \) is a conclusion of some \( A \in \text{Args} \). Then we will say that \( A \) (strictly) \( \text{Args} \)-defeats \( B \) on the basis of \( \Gamma \) if and only if \( A \) (strictly) defeats \( B \) on the basis of \( \Gamma', <_{\text{Args}} \). In the second case, the definition of (strict) defeat refers to the definition used in the framework for fixed priorities.

The definition of acceptability incorporates this notation. An argument \( A \) will be acceptable with respect to a set \( \text{Args} \) of arguments if and only if all arguments \( \text{Args} \)-defeating \( A \) are strictly \( \text{Args} \)-defeated by some argument in \( \text{Args} \).

To redefine the characteristic function of an ordered theory for handling priorities, Prakken and Sartor introduce some slight modifications, since monotonicity for the \( F_\Gamma \) operator is restricted for conflict-free sets of arguments.

If \( \Gamma = (S, D) \) is an ordered theory, \( S \subset \text{Args}_\Gamma \), and \( C_{\text{Args}_\Gamma} \) is the set of all conflict-free subsets of \( \text{Args}_\Gamma \), then the characteristic function of \( \Gamma \) is defined as

- \( G_\Gamma : C_{\text{Args}_\Gamma} \rightarrow \text{Pow}(\text{Args}_\Gamma) \)
- \( G_\Gamma(S) = \{ A \in \text{Args}_\Gamma \mid A \text{ is acceptable w.r.t. } S \} \)

where “acceptable” corresponds to the notion defined above in terms of \( \text{Args} \)-defeat.

The authors show that \( G_\Gamma \) is monotonic, and that the former concepts of justified, overruled and defeasible arguments can be defined analogously (e.g., an argument \( A \) is justified if and only if \( A \) is in the least fixpoint of \( G_\Gamma \)). It can be also shown that if \( \Gamma \) is finitary, by defining \( G^1 = G_\Gamma(\emptyset) \) and \( G^{i+1} = G^i \cup G_\Gamma(F^i) \), it holds that \( \bigcup_{i=0}^{\infty} (G^i) = \text{JustArgs}_\Gamma \).

Finally, Prakken and Sartor show how their dialectical proof theory can be properly extended to reason in an argumentation framework, with defeasible priorities. To illustrate their framework, Prakken and Sartor present the following example.

**Example 3.17** [Prakken and Sartor 1996b]. In Italy, according to town planning regulations, it is possible to identify a priority rule stating that rules intended to protect artistic buildings have a higher priority than rules concerning town planning. This priority rule, however, usually conflicts with the Lex Posterior principle, which states that the later rule has priority over the earlier one. In terms of Prakken and Sartor’s formalism, an instance of this problem can be written as follows:

- \( r_1(x) \): \( x \) is a protected building \( \Rightarrow \neg x \text{'s exterior may be modified} \)
- \( r_2(x) \): \( x \) needs restructuring \( \Rightarrow x \text{'s exterior may be modified} \)
- \( r_3(x, y) \): \( x \) is about the protection of artistic buildings \( \land y \) is a town planning rule \( \Rightarrow y < x \)
- \( r_4(r_1(x)) \): \( r_1(x) \) is about the protection of artistic buildings
- \( r_5(r_2(x)) \): \( r_2(x) \) is a town planning rule
- \( r_6(r_1(x)) \)
- \( r_7(x, y) \)
- \( r_8(Villa_0) \)
- \( r_{9}(T(x, y)) \)
- \( r_{9}(T(x, y)) \)

The dispute begins with the proponent’s first move, in which he asserts that \( Villa_0 \) is a protected building and therefore its exterior may not be modified. The opponent responds by asserting that \( Villa_0 \) needs restructuring and then its exterior may be modified. However, since the protection rule prevails over the town planning rule, the proponent can strictly defeat the argument advanced by the opponent. But this priority argument can be defeated by the opponent using a conflicting priority argument based on the Lex Posterior principle. The dispute goes on...
in an interesting way when the proponent takes the debate to the meta-metalevel and asserts that since the Lex Posterior principle is earlier than the building regulations principle, the former is inferior to the latter on the basis of the former.

This finishes the dispute, since the opponent has run out of moves. Hence the argument advanced by the proponent can be considered as a provably justified argument. The resulting debate is shown in Figure 5.

### 3.8 Verheij’s CUMULA Argumentation Model

Bart Verheij’s work [Verheij 1996] has two main purposes. The first purpose is to study the role of rules and reasons in argumentation, their relevant properties and characteristics, and the relationships among these properties. This gives rise to a model called Reason-Based Logic [Hage and Verheij 1995]. The second purpose is to study how the status of an argument is determined by its structure, the associated counterarguments, and the stage reached in an argumentation process. In order to achieve this purpose a second model of argumentation called CUMULA is proposed.

Reason-Based Logic is a theory of rules and reasons that is based on an extended, FOL-like language with special function and predicate symbols for translating natural language sentences to terms. Thus, it is possible to state facts such as \( \text{Rule}(\text{weather report}(X), X) \), where \( X \) is a variable representing some type of weather. Rules are objects that represent a relation between a condition and a conclusion. The fact that a rule is valid is expressed by the sentence

\[
\text{Valid}(\text{rule}(\text{weather report}(X), X))
\]

According to Verheij, rules give rise to the reasons that are used in arguments to support a conclusion. A rule gives rise to a reason if the rule applies. From the fact that \( \text{weather report(rainy day)} \), to state that a rule applies:

\[
\text{Applies}(\text{rule}(\text{weather report}(X), X)), \text{weather report(rainy day)}, \text{rainy day})
\]

Sometimes, although the conditions of a rule holds, its conclusion does not follow. This is because of the existence of exclusionary reasons, which are reasons that make a rule inapplicable.

\[
\text{Reason(bad local prediction), excluded(rule(\text{weather report}(X),X) weather report(rainy day), rainy day))}
\]

From the fact \( \text{bad local prediction} \), it can be inferred that the rule \( \text{rule(weather report}(X), X) \) is excluded, and no longer applicable.

Reasons for and against a conclusion can be weighed. For instance, we may have

\[
\text{Reason(local prediction, rainy day) and Reason(personal belief, \lnot rainy day).}
\]

In such a case, whether a conclusion follows depends on how the exclusionary
reasons are weighed. The outcome of the weighing can be modified if new reasons are taken into account. Reasons are also needed at the moment of applying a rule (i.e., the act of applying a rule depends on the existence of reasons for performing such an act).

Verheij proposes a formalization of the concepts mentioned above, presenting an extended, generalized definition of deduction. The exclusion of rules when exclusionary reasons apply results in a type of nonmonotonic reasoning. A distinction between reasoning with rules and reasoning with principles is drawn. Although both rules and principles are used to represent connections between a condition and a conclusion, there are some slight differences, the most important one being that principles are dependent on other rules and principles, while rules are not. At the same time there is a basic relationship between rules and principles: every rule has underlying principles.

Cumulative Argumentation, or CUMULA, is a formal model of argumentation in stages. Arguments are seen as treelike structures of sentences describing the way a conclusion is supported. The simplest type of argument with non-trivial structure is the single-step argument, which has the form “Reason. So Conclusion.” A reason in an argument can consist of several subreasons, which schematically results in the form “Subreason\(_1\), \ldots, Subreason\(_n\). So Conclusion.” Arguments can be combined by subordination or by coordination. Two arguments are combined by subordination if the conclusion \(h\) of one argument is the (sub)reason of a single-step argument. Two arguments are combined by coordination if both of them support the same conclusion \(h\). In this case, we get a stronger support for \(h\).

**Example 3.18** [Verheij 1996]. The following is a single-step argument:

\[
\text{The sun is shining. So, it is a beautiful day}
\]

where \textit{The sun is shining} is the reason, and \textit{it is a beautiful day} is the conclusion. The argument

\[
\text{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat.}
\]

This argument results from the subordination of the arguments \textit{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat}.

Given the arguments \textit{The sun is shining. So, it is a beautiful day} and \textit{The sun is shining. So, it is a beautiful day}. Coordinating these two arguments, we obtain

\[
\text{The sun is shining; the sky is blue. So, it is a beautiful day}
\]

In the same way as Pollock [1987], Verheij distinguishes between undercutting and rebutting defeat, but he discusses additional kinds of defeat, namely defeat by sequential weakening, which refers to the fact that arguments cannot be indefinitely chained, and defeat by parallel strengthening, which refers to the case in which the coordination of two arguments defeats one of the two arguments. More general kinds of defeat occur when there is a group of arguments whose conclusions cannot hold simultaneously. Collective defeat refers to a form of defeat in which a group of arguments is defeated as a whole, while each argument that comprises the group is not individually defeated. Indeterministic defeat occurs when there are several choices of arguments that can be made and the choice is indeterministically solved. It is necessary to take one of the arguments as defeated in order to support the others.

**Example 3.19** [Verheij 1996] (different types of defeat). Next we will introduce different types of defeat characterized within Verheij’s approach.

- (Undercutting defeat) Consider the following single-step argument (from Pollock): \textit{The object looks red. So, the object is red}. Suppose we also have the following argument: \textit{The object is illuminated by a red light. Taking both arguments into account, the argument

\[
\text{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat.}
\]

This argument results from the subordination of the arguments \textit{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat}.

Given the arguments \textit{The sun is shining. So, it is a beautiful day} and \textit{The sun is shining. So, it is a beautiful day}. Coordinating these two arguments, we obtain

\[
\text{The sun is shining; the sky is blue. So, it is a beautiful day}
\]

In the same way as Pollock [1987], Verheij distinguishes between undercutting and rebutting defeat, but he discusses additional kinds of defeat, namely defeat by sequential weakening, which refers to the fact that arguments cannot be indefinitely chained, and defeat by parallel strengthening, which refers to the case in which the coordination of two arguments defeats one of the two arguments. More general kinds of defeat occur when there is a group of arguments whose conclusions cannot hold simultaneously. Collective defeat refers to a form of defeat in which a group of arguments is defeated as a whole, while each argument that comprises the group is not individually defeated. Indeterministic defeat occurs when there are several choices of arguments that can be made and the choice is indeterministically solved. It is necessary to take one of the arguments as defeated in order to support the others.

**Example 3.19** [Verheij 1996] (different types of defeat). Next we will introduce different types of defeat characterized within Verheij’s approach.

- (Undercutting defeat) Consider the following single-step argument (from Pollock): \textit{The object looks red. So, the object is red}. Suppose we also have the following argument: \textit{The object is illuminated by a red light. Taking both arguments into account, the argument

\[
\text{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat.}
\]

This argument results from the subordination of the arguments \textit{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat}.

Given the arguments \textit{The sun is shining. So, it is a beautiful day} and \textit{The sun is shining. So, it is a beautiful day}. Coordinating these two arguments, we obtain

\[
\text{The sun is shining; the sky is blue. So, it is a beautiful day}
\]

In the same way as Pollock [1987], Verheij distinguishes between undercutting and rebutting defeat, but he discusses additional kinds of defeat, namely defeat by sequential weakening, which refers to the fact that arguments cannot be indefinitely chained, and defeat by parallel strengthening, which refers to the case in which the coordination of two arguments defeats one of the two arguments. More general kinds of defeat occur when there is a group of arguments whose conclusions cannot hold simultaneously. Collective defeat refers to a form of defeat in which a group of arguments is defeated as a whole, while each argument that comprises the group is not individually defeated. Indeterministic defeat occurs when there are several choices of arguments that can be made and the choice is indeterministically solved. It is necessary to take one of the arguments as defeated in order to support the others.

**Example 3.19** [Verheij 1996] (different types of defeat). Next we will introduce different types of defeat characterized within Verheij’s approach.

- (Undercutting defeat) Consider the following single-step argument (from Pollock): \textit{The object looks red. So, the object is red}. Suppose we also have the following argument: \textit{The object is illuminated by a red light. Taking both arguments into account, the argument

\[
\text{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat.}
\]

This argument results from the subordination of the arguments \textit{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat}.

Given the arguments \textit{The sun is shining. So, it is a beautiful day} and \textit{The sun is shining. So, it is a beautiful day}. Coordinating these two arguments, we obtain

\[
\text{The sun is shining; the sky is blue. So, it is a beautiful day}
\]

In the same way as Pollock [1987], Verheij distinguishes between undercutting and rebutting defeat, but he discusses additional kinds of defeat, namely defeat by sequential weakening, which refers to the fact that arguments cannot be indefinitely chained, and defeat by parallel strengthening, which refers to the case in which the coordination of two arguments defeats one of the two arguments. More general kinds of defeat occur when there is a group of arguments whose conclusions cannot hold simultaneously. Collective defeat refers to a form of defeat in which a group of arguments is defeated as a whole, while each argument that comprises the group is not individually defeated. Indeterministic defeat occurs when there are several choices of arguments that can be made and the choice is indeterministically solved. It is necessary to take one of the arguments as defeated in order to support the others.

**Example 3.19** [Verheij 1996] (different types of defeat). Next we will introduce different types of defeat characterized within Verheij’s approach.

- (Undercutting defeat) Consider the following single-step argument (from Pollock): \textit{The object looks red. So, the object is red}. Suppose we also have the following argument: \textit{The object is illuminated by a red light. Taking both arguments into account, the argument

\[
\text{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat.}
\]

This argument results from the subordination of the arguments \textit{A colleague is completely soaked and tells that it is raining. So, it is probably raining. So, it is wise to put on a raincoat}.

Given the arguments \textit{The sun is shining. So, it is a beautiful day} and \textit{The sun is shining. So, it is a beautiful day}. Coordinating these two arguments, we obtain

\[
\text{The sun is shining; the sky is blue. So, it is a beautiful day}
\]

In the same way as Pollock [1987], Verheij distinguishes between undercutting and rebutting defeat, but he discusses additional kinds of defeat, namely defeat by sequential weakening, which refers to the fact that arguments cannot be indefinitely chained, and defeat by parallel strengthening, which refers to the case in which the coordination of two arguments defeats one of the two arguments. More general kinds of defeat occur when there is a group of arguments whose conclusions cannot hold simultaneously. Collective defeat refers to a form of defeat in which a group of arguments is defeated as a whole, while each argument that comprises the group is not individually defeated. Indeterministic defeat occurs when there are several choices of arguments that can be made and the choice is indeterministically solved. It is necessary to take one of the arguments as defeated in order to support the others.
that the object is red does not justify its conclusion. In this case, the fact that the object is illuminated by a red light undercuts the argument that the object is red.

- **(Rebutting defeat)** Consider then the following two arguments: John likes French fries. So, he orders French fries and John is on a low-calorie diet. So he does not order French fries. These arguments support opposite conclusions, rebutting each other.

- **(Sequential weakening)** Consider the following argument:

  This body of grains of sand is a heap.  
  So, this body of grains of sand minus 1 grain is a heap.  
  So, this body of grains of sand minus 2 grains is a heap.  
  \ldots  
  So, this body of grains of sand minus n grains is a heap.

  In this example, each single step of the argument is correct, but clearly the argument cannot be pursued indefinitely, since in the end there is no grain of sand left. The important point here is that it is impossible to choose a single step that makes the argument defeated. Only because the step is repeated too often is, the argument is weakened below the limit of acceptability and defeated. In such a case we speak of defeat by sequential weakening of the argument.

- **(Parallel strengthening)** Assume that John has committed an offense, but is a minor first offender. As a result, the judge might consider the following argument:

  John is a minor first offender. So, John should not be punished. (1)  

  If John has robbed Alex, the judge might consider this an argument that rebuts the following argument with opposite conclusion:

  John has robbed Alex. So, John should be punished. (2)  

  In the case of rebuttal the judge decided not to punish John. The judge might decide analogously if John has injured Alex in a fight. Nevertheless, if John has both robbed Alex and injured him in a fight, the judge might decide differently. Since there are now two reasons for punishing John, coordination of arguments gives us the following composite argument:

  John has robbed Alex; John injured Alex in a fight. (3)  
  So, John should be punished.

  This new argument might defeat the argument that John should not be punished. In that case, argument (3) defeats argument (2), whereas argument (1) does not. In this example the defeat of the argument not to punish John can be explained by the parallel strengthening of the argument to punish him.

- **(Collective and indeterministic defeat)** Consider an employer who wants to hire two persons if they are qualified. If John, Alex, and Mary are qualified, the employer can make the following arguments:

  John is qualified. So, John is hired (1)  
  Alex is qualified. So, Alex is hired (2)  
  Mary is qualified. So, Mary is hired (3)  

  Since the employer only wants to hire two persons, the three arguments cannot all be undefeated. If there is no additional information to resolve this conflict of arguments, two approaches can be distinguished to resolve the conflict: collective and indeterministic defeat.

  In the first approach, all arguments are considered defeated. We speak of collective defeat if a group of arguments is defeated as a whole, while the arguments in the group would not be defeated on their own. In this approach the set of arguments \{ (1), (2), (3) \} would be defeated.

  In the second approach, the conflict is resolved by considering one of the arguments in the conflict defeated. Since there are several choices that can be made, neither of them being better than the others, the conflict is indeterministically solved: each choice of a defeated argument is allowed. In this approach the arguments (1), (2), or (3) could be considered as defeated, resulting in the
sets of undefeated arguments \( \{ (2), (3) \}, \{ (1), (3) \} \) and \( \{ (1), (2) \} \), respectively.

An argument is considered to have two possible statuses: undefeated, if it justifies its conclusion, or defeated, if it does not justify its conclusion. A set of arguments \( \{ a_1, a_2, \ldots, a_k \} \) along with the associated defeat status for each \( a_i \) comprises a stage in the argumentation process. Starting from a given set of premises, arguments are constructed gradually in six different elementary ways: (a) introducing a new statement; (b) adding a forward step (the conclusion of an existing argument is used to support a new conclusion); (c) adding a backward step (the premise of an existing argument is replaced by one or more new premises); (d) adding a broadening step (the conclusion of an existing argument is supported by an additional reason); (e) subordinating one argument to another; (f) coordinating two arguments. These last two concepts were explained above.

The construction of new arguments from the arguments taken into account at some stage leads us to a new stage.

Example 3.20 [Verheij 1996]. Consider the statement \textit{It is raining}. It corresponds to an argument with trivial structure (case a). A forward step (case b) can be added by considering the fact that it is raining can be used to support whether to put on a raincoat or not. This results in the following single-step argument:

\begin{align*}
\text{It is raining.} \\
\text{So, it is wise to put on a raincoat.}
\end{align*}

A backward step (case c) can be added to an argument, meaning that the premise of the argument is supported by a new premise. For instance, suppose that we are in a room with no windows and a colleague comes in completely soaked and says it is raining. Thus we can build the argument:

\begin{align*}
\text{A colleague is completely soaked and says that it is raining.} \\
\text{So, it is raining.}
\end{align*}

The premise \textit{It is raining} can now be replaced by the new premise \textit{A colleague is completely soaked and says that it is raining}.

A broadening step (case d) can be added to an argument, meaning that the conclusion of a (nontrivial) argument is supported by an additional reason. Consider for instance the previous argument, which can be broadened to the following argument:

\begin{align*}
\text{A colleague is completely soaked and says that it is raining;} \\
\text{The weather report on the radio says that it is raining.} \\
\text{So, it is raining.}
\end{align*}

Two arguments can also be combined by subordination (case e) if one of the arguments taken into account has a premise that is the conclusion of the other. For instance the argument:

\begin{align*}
\text{A colleague is completely soaked and says that it is raining.} \\
\text{So, it is raining.}
\end{align*}

and

\begin{align*}
\text{A colleague is completely soaked and says that it is raining.} \\
\text{So, it is raining.}
\end{align*}

Two arguments can also be combined by coordination (case f) if they have the same conclusion. For instance, the arguments

\begin{align*}
\text{A colleague is completely soaked and says that it is raining.} \\
\text{So, it is raining.}
\end{align*}

and

\begin{align*}
\text{The weather report on the radio says that it is raining.} \\
\text{So, it is raining.}
\end{align*}

Finally, two arguments can be combined by coordination (case f) if they have the same conclusion. For instance, the arguments

\begin{align*}
\text{A colleague is completely soaked and says that it is raining.} \\
\text{So, it is raining.}
\end{align*}

and

\begin{align*}
\text{A colleague is completely soaked and says that it is raining.} \\
\text{The weather report on the radio says that it is raining.} \\
\text{So, it is raining.}
\end{align*}
Lines of argumentation are defined as sequences of argument stages. The status of an argument may change from stage to stage, so it changes along the line of argumentation. Although Verheij provides a construction of the arguments in the ranges of the successor stages, he does not discuss the statuses of these arguments. He argues that “it is probably not easy to define the relation between the statuses of the arguments in the range of a stage and in the range of a successor, since a change of status of one argument can affect the status of a cascade of other arguments.” In this respect, his approach seems weaker than Vreeswijk’s or Simari–Loui’s, where a theory of justification is defined.

3.9 Bondarenko, Dung, Kowalski, and Toni’s Abstract Argumentation Framework

The authors develop an abstract assumption-based framework for default reasoning, combining the semantics of logic programming [Bondarenko et al. 1997] and the notion of extension as presented in Poole’s THEORIST [Poole 1988]. In their framework, assumptions can be defeated or “attacked” if the contrary can be proved, possibly with the aid of other conflicting assumptions. The framework can be understood as a generalization of THEORIST that allows any theory formulated in a monotonic logic to be extended by a defeasible set of assumptions. The authors do not model argument as a phenomenon, but instead use the idea of argument to understand logic programming.

The assumption-based framework departs from a monotonic deductive system, which is composed of a formal language $\mathcal{L}$ and a set of inference rules $\mathcal{R}$. Following Poole [1988], the authors argue that the nonmonotonic character of default reasoning arises because a set of assumptions that acceptably extends a given theory in a monotonic logic might not be acceptable if new sentences are added to the theory. Different logics for default reasoning can be seen as having different underlying monotonic logics, different kinds of assumptions, and different notions for acceptability.

An assumption-based framework is intended to capture all these features, by sanctioning when a set of assumptions can be accepted as an extension for a given theory.

Formally, an assumption-based framework is a triple $(T, Ab, \bot)$, where $T, Ab \subseteq \mathcal{L}$, $Ab \neq \emptyset$, and $\bot$ is a mapping from $Ab$ into $\mathcal{L}$, where $\overline{\alpha}$ denotes the contrary of $\alpha$. A theory expresses a set of beliefs, while $Ab$ accounts for a set of assumptions that can be used to extend $T$.

Given a theory $T$ and a set of assumptions $\Delta \subseteq Ab$, we can form the deductive closure $Th(T \cup \Delta)$, often called an extension in the literature on nonmonotonic logic. The goal is to define which constraints make an extension acceptable.

3.9.1 Naive Semantics. The simplest extension responds to what is called the naive semantics. The only requirement in this case is that the extensions must be maximal and conflict-free. Given an assumption-based framework $(T, Ab, \bot)$, $\Delta \subseteq Ab$ is conflict-free if and only if $\Delta$ is such that for all $\alpha \in Ab$, $T \cup \Delta \not\vdash \alpha, \overline{\alpha}$.

The authors show that the naive semantics is guaranteed to exist for assumption-based frameworks that admit at least one conflict-free extension. They also define several systems (such as THEORIST [Poole 1988], circumscription [McCarthy 1980], logic programming, default logic [Reiter 1980], and autoepistemic logic [Moore 1985]) in terms of an assumption-based frameworks.

**Example 3.21** [Bondarenko et al. 1997]. We will consider normal logic programs as a particular instance of assumption-based frameworks. As a basis we will use a first-order language $\mathcal{L}$, writing $\mathcal{H}B$ to denote the Herbrand base (i.e., the set of all ground atoms in $\mathcal{L}$ formulated in terms of the Herbrand universe). We will write $\mathcal{H}B_{\text{not}}$ for denoting the set $\{ \alpha \mid \alpha \in \mathcal{H}B \}$. The set of all literals $\text{Lit}$ in $\mathcal{L}$ will be $\mathcal{H}B \cup \mathcal{H}B_{\text{not}}$.

A normal logic program $T$ is a set of clauses of the form $\alpha \leftarrow \beta_1, \ldots, \beta_n$, where $\alpha \in \mathcal{H}B$, $\beta_1, \ldots, \beta_n \in \text{Lit}$, and $n \geq 0$.  

The assumption-based framework corresponding to such a normal program \( T \) is \((T, \mathcal{H}B_{not}, \sim)\) with respect to \((\mathcal{L}, \mathcal{R})\), where

- \( \mathcal{L} = \text{Lit} \cup \{ \alpha \leftarrow \beta_1, \ldots, \beta_n \mid \alpha \in \mathcal{H}B \} \), and \( \beta_1, \ldots, \beta_n \in \text{Lit} \), and \( n \geq 0 \).
- \( \mathcal{R} \) is the set of all inference rules of the form
  
  \[
  \alpha \leftarrow \beta_1 \ldots \beta_n \quad \beta_1, \ldots, \beta_n \quad \alpha
  \]

  where \( \alpha \in \mathcal{H}B \), and \( \beta_1, \ldots, \beta_n \in \text{Lit} \), and \( n \geq 0 \);
- \( \lnot \alpha = \alpha \), for each not \( \alpha \in \mathcal{H}B_{not} \).

3.9.2 Stable Semantics. The second, credulous semantics, called \textit{stable semantics}, generalizes the stable model semantics of logic programming, as well as the standard semantics of default logic, autoepistemic logic, and nonmonotonic modal logic. Informally, a set of assumptions is \textit{stable} if it is conflict-free and it attacks every assumption it does not contain. Given \( \Delta \subseteq \text{Ab} \), we say that \( \Delta \) \textit{attacks} \( \alpha \in \text{Ab} \) if and only if \( T \cup \Delta \vdash \lnot \alpha \). If \( \Delta \subseteq \text{Ab} \) is such that \( \Delta = \{ \alpha \in \text{Ab} \mid T \cup \Delta \vdash \lnot \alpha \} \), then \( \Delta \) is said to be \textit{closed}.

Formally, a set of assumptions \( \Delta \) is \textit{stable} if and only if

a. \( \Delta \) is closed,

b. \( \Delta \) does not attack itself, and

c. \( \Delta \) attacks each assumption \( \alpha \notin \Delta \).

The stable semantics generalizes the naive semantics, in the sense that for any assumption-based framework \( (T, \text{Ab}, \sim) \), for any set of assumptions \( \Delta \subseteq \text{Ab} \), if \( \Delta \) is stable then \( \Delta \) is maximally conflict-free.

\textbf{Example 3.22} As stated in Example 3.21, in the assumption-based framework for a normal logic program \( T \) the set of assumptions \( \text{Ab} \) is defined by \( \mathcal{H}B_{not} \). In this case, the attack relation is defined as follows: \( \Delta_1 \subseteq \text{Ab} \) attacks \( \text{not} \ p \) if \( T \cup \Delta_1 \vdash \lnot p \).

Consider the normal logic program \( P \) defined as follows:

\[
\{ s \leftarrow r, r \leftarrow \text{not} \ s \}
\]

Then \( \Delta_1 \) is stable; since it is closed, it does not attack itself (i.e., \( T \cup \Delta_1 \not\vdash \lnot s \)) and it attacks every assumption it does not contain (i.e., \( T \cup \Delta_1 \vdash r \)). The same holds for \( \Delta_2 \). However, \( \Delta_1 \cup \Delta_2 \) is not stable (since it attacks itself).

A framework \( (T, \text{Ab}, \sim) \) is called \textit{normal} if and only if every maximal conflict-free set of assumptions is stable. The authors also show that the notion of stable extension can be characterized through alternative definitions, which basically differ from each other in the way they characterize theoremhood in the underlying monotonic logic.

Assumption-based frameworks in which every set of assumptions \( \Delta \subseteq \text{Ab} \) is closed deserve special consideration. Such frameworks are said to be \textit{flat}. In flat frameworks, assumptions can be seen as independent from one another.

3.9.3 Admissibility and Preferential Semantics. The third, credulous semantics, \textit{admissibility semantics}, generalizes admissibility semantics in logic programming [Dung 1995b]. This semantics is not as liberal as the naive semantics, but not as restrictive as the stable one. The difference between a stable extension and an admissible extension is that instead of attacking every assumption that it does not contain, the latter attacks every assumption that attacks it. Formally, a closed set of assumptions \( \Delta \subseteq \text{Ab} \) is \textit{admissible} if and only if (a) \( \Delta \) does not attack itself and (b) for each closed set of assumptions \( \Delta' \subseteq \text{Ab} \), if \( \Delta' \) attacks \( \Delta \), then \( \Delta \) attacks \( \Delta' \).

\textbf{Example 3.23} [Bondarenko et al. 1997]. Consider the logic program \( P = \{ p \leftarrow \text{not} \ p \} \). It has no stable extensions. However, \( \Delta = \emptyset \) is admissible, since it is conflict-free and it is not attacked by any other set of assumptions. Moreover, \( \Delta \) is maximally admissible, because the only larger set \( \text{not} \ p \) attacks itself.

The fourth, credulous semantics, called \textit{preferential semantics}, simply regards an extension as acceptable if it is maximally
admissible, in the sense that no proper subset of the extension is also admissible.

It can be shown that for normal assumption-based frameworks, the notions of maximal conflict-free, stability, and being preferred are equivalent.

**Example 3.24** [Bondarenko et al. 1997]. In the assumption-based framework corresponding to the logic program \{ p \leftarrow \lnot q, q \leftarrow \lnot r, r \leftarrow \lnot p \} there is only one preferred set of assumptions, namely \( \emptyset \), which is not maximally conflict-free. In fact, the maximally conflict-free sets of assumptions are \{ \lnot p \}, \{ \lnot q \} and \{ \lnot r \}, which are not admissible.

### 3.9.4 Complete and Skeptical Semantics

Admissible extensions such that they contain all the assumptions they defend are called complete extensions, which give rise to the characterization of a fifth, credulous semantics, called complete semantics, which is intermediate between the admissibility and preferential semantics. It regards an extension as acceptable if it is admissible and it contains all assumptions it defends.

Formally, a set of assumptions \( \Delta \) defends an assumption \( \alpha \) if and only if for each closed set of assumptions \( \Delta' \), if \( \Delta' \) attacks \( \alpha \) then \( \Delta \) attacks \( \Delta' - \Delta \). From this definition, it can be shown that admissibility of a set of assumptions \( \Delta \) is equivalent to \( \Delta \) being closed, such that \( \Delta \subseteq \text{Def}(\Delta) \), where \( \text{Def}(\Delta) = \{ \alpha \mid \Delta \text{ defends } \alpha \} \).

A set of assumptions \( \Delta \) is said to be complete if and only if it is identical to the set of assumptions it defends; that is, \( \Delta \) is closed and it holds that \( \Delta = \text{Def}(\Delta) \). The authors show that every stable set of assumptions is complete.

**Example 3.25** [Bondarenko et al. 1997]. Let \( T \) be the logic program

\[
\{ p \leftarrow \lnot q, q \leftarrow \lnot p, r \leftarrow p, r \leftarrow q \}
\]

There are two stable sets of assumptions \{ \lnot p \} and \{ \lnot q \}, which are also complete.

Each of the above semantics has a credulous and a skeptical version. The former semantics (naive, stable, admissible, and complete) were credulous. According to the credulous approach, a conclusion holds if there is at least one acceptable extension from which a conclusion can be obtained by means of the underlying monotonic logic. The skeptical approach justifies only those conclusions that can be derived in all acceptable extensions.

The semantics of several approaches to default reasoning can be expressed as instances of the given abstract framework. By taking the underlying monotonic logic as first order classical logic, the credulous version of the naive semantics generalizes THEORIST semantics. On the contrary, circumscription can be understood as the skeptical version of the naive semantics, also using first-order logic as the underlying monotonic logic. On the other hand, the credulous version of stable semantics generalizes the stable model semantics of logic programming as well as the semantics of some consistency-based approaches to default reasoning such as default logic, non-monotonic logic, and autoepistemic logic. The authors also show that by taking the underlying monotonic logic as the logic of Horn clauses, the skeptical version of the complete semantics gives rise to the well-founded semantics of logic programming.

**Example 3.26** Consider the logic program \( T \) in Example 3.25. Its well-founded semantics \( \text{Wfs}(T) \) can be obtained by intersecting all complete sets of assumptions; that is, \( \text{Wfs}(T) = \{ \lnot p \} \cap \{ \lnot q \} = \emptyset \).

In recent work [Kakas and Toni 1999] the abstract argumentation framework has been used as a basis for defining an unifying proof theory for various argumentation semantics of logic programming. The proof theory is based on the derivation of trees, where each node in a tree is an argument. The proposed approach generalizes directly to other logics for nonmonotonic reasoning.

### 4. CONCLUSIONS

We have discussed the most relevant ideas concerning models of argumentation. The
approaches developed so far have contributed to a better understanding of reasoning problems in AI, leading to new, alternative formalizations. Areas in which argumentation has proven useful include case-based reasoning, temporal reasoning, natural language understanding, AI and law, logic programming, negotiation, and hypertext systems, among others.

In recent years, argumentation has been found to be particularly powerful in the areas of knowledge representation, commonsense reasoning, logic programming, legal reasoning, decision, and negotiation. Argumentation within a logic-programming setting has provided extensions for the semantics of logic programming [Bondarenko et al. 1997; Dung 1993b; Kowalski and Toni 1996]. A new understanding of dialectics has contributed to knowledge representation and reasoning formalisms for legal applications in AI, and provide outstanding examples of successful new ideas from AI that could revise both the philosophy of the field and practice. Recent research in this area has also led to finding common foundations for extended logic programming and procedural argumentative frameworks [Prakken and Sartor 1996b; 1997]. Negotiation and argument games [Loui 1997; Loui et al. 1997] have opened a new way of conceptualizing computation, posing again old questions concerning judgment, agreement, decision making, and social choice.

Prior to the research on argumentation developed in this last decade, demonstrative, nonampliative reasoning had been the basis of formal languages for modeling computation, whereas nondemonstrative, ampliative reasoning was only confined within special informal linguistic systems. Argumentation presents itself as the natural tool for characterizing several kinds of ampliative reasoning, being thus an attractive approach to that major goal in AI: modeling commonsense reasoning. Though much has been achieved, the most promising results seem to be still ahead.

ACKNOWLEDGMENTS

We want to thank anonymous referees for suggestions and comments that helped to improve the structure of the first version of this paper.

REFERENCES


BARTH, E. M. AND KRABBE, E. C. W. 1982. From Axiom to Dialogue: A Philosophical Study of
Logical Models of Argument


Cafrisy, R. 1990. EVID: A System for Interactive Defeasible Reasoning. Artificial Intelligence Laboratory, University of Texas at Austin.


DUNG, P. M. 1995a. An argumentation theoretic


DELOMBANE, J. 1986. A propositional logic for natural


Logical Models of Argument


Logical Models of Argument


Poole, D. L. 1985. On the comparison of theories: preferring the most specific explanation. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 144–147. IJCAI.


reasoning. *Artificial Intelligence and Law* 4, 331–368.


WALTON, D. N. 1990. What is reasoning? What is an argument? The Journal of Philosophy 87, 8 (Aug.).


Received April 1998; revised May 1999; accepted October 1999