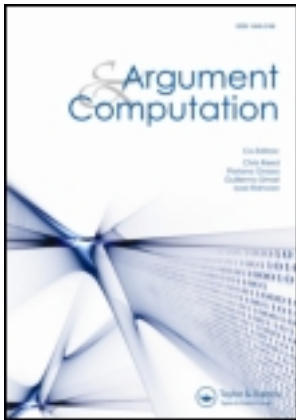


This article was downloaded by: [94.8.183.5]

On: 26 February 2014, At: 11:23

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



## Argument & Computation

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/tarc20>

### Introduction to structured argumentation

Philippe Besnard<sup>a</sup>, Alejandro Garcia<sup>b</sup>, Anthony Hunter<sup>c</sup>, Sanjay Modgil<sup>d</sup>, Henry Prakken<sup>ef</sup>, Guillermo Simari<sup>b</sup> & Francesca Toni<sup>g</sup>

<sup>a</sup> IRIT, Université Paul Sabatier, Toulouse, France

<sup>b</sup> Department of Computer Science & Engineering, Universidad Nacional del Sur, Bahia Blanca, Argentina

<sup>c</sup> Department of Computer Science, University College London, London, UK

<sup>d</sup> Department of Informatics, Kings College London, London, UK

<sup>e</sup> Department of Information and Computing Sciences, Universiteit Utrecht, Utrecht, The Netherlands

<sup>f</sup> Faculty of Law, University of Groningen, Groningen, The Netherlands

<sup>g</sup> Department of Computing, Imperial College London, London, UK  
Published online: 11 Feb 2014.

To cite this article: Philippe Besnard, Alejandro Garcia, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Simari & Francesca Toni (2014) Introduction to structured argumentation, *Argument & Computation*, 5:1, 1-4, DOI: [10.1080/19462166.2013.869764](https://doi.org/10.1080/19462166.2013.869764)

To link to this article: <http://dx.doi.org/10.1080/19462166.2013.869764>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

## Introduction to structured argumentation

Philippe Besnard<sup>a</sup>, Alejandro Garcia<sup>b</sup>, Anthony Hunter<sup>c\*</sup>, Sanjay Modgil<sup>d</sup>, Henry Prakken<sup>e,f</sup>,  
Guillermo Simari<sup>b</sup> and Francesca Toni<sup>g</sup>

<sup>a</sup>IRIT, Université Paul Sabatier, Toulouse, France; <sup>b</sup>Department of Computer Science & Engineering, Universidad Nacional del Sur, Bahía Blanca, Argentina; <sup>c</sup>Department of Computer Science, University College London, London, UK; <sup>d</sup>Department of Informatics, Kings College London, London, UK; <sup>e</sup>Department of Information and Computing Sciences, Universiteit Utrecht, Utrecht, The Netherlands; <sup>f</sup>Faculty of Law, University of Groningen, Groningen, The Netherlands; <sup>g</sup>Department of Computing, Imperial College London, London, UK

(Received 1 November 2013; final version received 19 November 2013)

In abstract argumentation, each argument is regarded as atomic. There is no internal structure to an argument. Also, there is no specification of what is an argument or an attack. They are assumed to be given. This abstract perspective provides many advantages for studying the nature of argumentation, but it does not cover all our needs for understanding argumentation or for building tools for supporting or undertaking argumentation. If we want a more detailed formalisation of arguments than is available with abstract argumentation, we can turn to structured argumentation, which is the topic of this special issue of *Argument and Computation*. In structured argumentation, we assume a formal language for representing knowledge, and specifying how arguments and counterarguments can be constructed from that knowledge. An argument is then said to be structured in the sense that normally the premises and claim of the argument are made explicit, and the relationship between the premises and claim is formally defined (for instance using logical entailment). In this introduction, we provide a brief overview of the approaches covered in this special issue on structured argumentation.

**Keywords:** formal models of argumentation; logic; argument in logic programming; automated argumentation reasoning systems < argument and automated reasoning

### 1. Introduction

Argumentation is an important cognitive capacity for handling conflicting information, viewpoints, opinions, etc. This has led to a number of interesting lines of research in artificial intelligence, and allied fields, leading to the emergence of computational models of argument as a promising research field.

Abstract argumentation is now established as a formalism that provides many important insights into the nature of argumentation. A situation involving argumentation can be represented by a directed graph. Each node represents an argument, and each arc denotes an attack by one argument on another. Such a graph can then be analysed to determine which arguments are acceptable according to some general criteria. This was originally proposed in the seminal paper by [Dung \(1995\)](#), and it has led to a large number of papers that have explored properties and developments of it.

In abstract argumentation, each argument is regarded as atomic. There is no internal structure to an argument. Also, there is no specification of what is an argument or an attack. They are assumed to be given. This abstract perspective provides many advantages for studying the nature

---

\*Corresponding author. Email: [a.hunter@cs.ucl.ac.uk](mailto:a.hunter@cs.ucl.ac.uk)

of argumentation, but it does not cover all our needs for understanding argumentation or for building tools for supporting or undertaking argumentation.

If we want a more detailed formalisation of arguments than is available with abstract argumentation, we can turn to *structured argumentation*, which is the topic of this special issue of *Argument and Computation*. In structured argumentation, we assume a formal language for representing knowledge, and specifying how arguments and counterarguments can be constructed from that knowledge. An argument is then said to be structured in the sense that normally the premises and claim of the argument are made explicit, and the relationship between the premises and claim is formally defined (for instance using logical entailment). This means we can describe arguments and attacks in structured argumentation as follows:

- *Argument* is a tuple containing a delineation of the support (premises) for the argument and the claim (conclusion) of the argument. There may also be other information associated with the argument such as how the claim is obtained from the support.
- *Attack* is a binary relation over arguments that denotes when one argument is in conflict with another argument. In structured argumentation, there is normally a formal definition for determining when this relation holds.

In addition, some approaches make a distinction between attack and defeat relations between arguments. In these works, preferences are applied to arguments, so that an attack only succeeds as a defeat if the attacked argument is not preferred to the attacking argument. Then, it is this defeat relation which is used as the binary relation in the Dung framework.

In the study of computational models of argument, there is a useful dichotomy that we can capture using the notions of monological argumentation and dialogical argumentation, as defined next.

- *Monological argumentation*: Here the emphasis is on the analysis of set of arguments and counterarguments that might be generated from some knowledge, or given by a third party. The aim is to evaluate them, determine what arguments are acceptable, draw conclusions from them, and perhaps make decisions based on them.
- *Dialogical argumentation*: Here the emphasis is on the exchange of arguments and counterarguments between agents. There is the consideration of how arguments and counterarguments are generated and evaluated, but there is also consideration of how the agents interact (i.e. what kinds of dialogical moves they can make), and of the options for retracting arguments, changing beliefs, etc. So dialogical argumentation includes consideration of protocols and strategies for the participants to follow.

In this special issue, we focus on monological argumentation. However, the approaches we present can be embedded in dialogical argumentation systems, so that the special issue also provides a grounding for an important aspect of dialogical argumentation.

In the following four papers in this special issue, we will consider four approaches to structured argumentation, namely ABA, ASPIC+, Defeasible Logic Programming (DeLP), and deductive argumentation. The primary intended audience for this special issue are graduate and undergraduate students, as well as researchers who want to use argumentation in their research, and who do not yet know (much) about formal and computational models of argumentation. Therefore, the papers in this issue will focus less on theoretical aspects and problems and more on design choices, examples and applications. However, we believe that researchers already familiar with the field will also benefit from this special issue, for example, to obtain a clearer understanding of how these approaches relate and differ and of how they can be applied in practice.

All four approaches in this special issue use logic as part of the formalisation, and there are numerous commonalities that exist between pairs of systems. However, there are also some interesting differences between the systems. We summarise each of these four systems as follows.

- *ABA*: ABA is a general framework for logic-based instantiation of abstract argumentation. A specific system in ABA is based on a deductive system, including a set of inference rules. A subset of the language in the deductive system is specified as assumptions. Each argument corresponds to a set of assumptions that, with the inference rules, proves a claim. Attacks between arguments are obtained in ABA via a notion of ‘contrary’ assumptions, specifying how to contradict them. Then, an argument attacks another if its claim contradicts (is the contrary of) some assumption in the other. ABA is equipped with a family of computational mechanisms for capturing specific argumentation semantics, corresponding to semantics for abstract argumentation.
- *ASPIC+*: The ASPIC+ framework is based on two ideas: the first is that conflicts between arguments are often resolved with explicit preferences, and the second is that arguments are built with two kinds of inference rules: strict, or deductive rules, whose premises guarantee their conclusion, and defeasible rules, whose premises only create a presumption in favour of their conclusion. The second idea implies that ASPIC+ does not primarily see argumentation as inconsistency handling in a given ‘base’ logic: conflicts between arguments may not only arise from the inconsistency of a knowledge base but also from the defeasibility of the reasoning steps in an argument. Accordingly, arguments can in ASPIC+ be attacked in three ways: on their uncertain premises or on their defeasible inferences, and the latter by either attacking their conclusion or the inference itself. ASPIC+ is not a system but a framework for specifying systems. A main objective is to identify conditions under which instantiations of ASPIC+ satisfy logical consistency and closure properties.
- *DeLP*: The DeLP system is based on the language of logic programming. It includes strong negation and default negation. The language was extended with the addition of defeasible rules and presumptions (that represent defeasible assumptions). Its inference engine relies on the construction of arguments to support literals, and the dialectical analysis of the reasons for and against accepting a particular argument. A DeLP argument, supporting a claim in the form of a literal, is considered a *warrant* for that claim if all the arguments that can be posed against it are *defeated*. Defeat comes in the form of an undefeated argument that *attacks* the original argument and it is also considered *better* (using some particular comparison criterion) than the one supporting the original claim. Attacks can be directed to the claim, on internal points of an argument, or its premises if they are presumptions of any form. The full analysis leads to the construction of a dialectical tree with the original argument in the root and whose nodes are arguments that can be marked defeated or undefeated, thus determining if the root argument is a warrant or not. Four possible answers are possible for a query: Yes, if the posed query is warranted; No, if the complement of the query is warranted; Undecided, if neither the query nor its complement is warranted; and Unknown, when the query is not in the signature of the program.
- *Deductive argumentation*: This is a logic-based approach to structured argumentation. Each argument is a pair  $(X, p)$  where  $X$  is a set of logical formulae, called the premises, that entails  $p$ , the claim, where entailment is specified by the choice of base logic (e.g. classical logic, modal logic, description logic, temporal logic, conditional logic, etc). Various options are available for specifying when one argument attacks another (e.g. argument A rebuts argument B when the claim of A is the negation of the claim of B). If we are then to construct an argument graph based on the arguments that can be constructed from a knowledgebase, there are various options available for selecting which arguments and attacks to include in

the argument graph (e.g. being exhaustive in including all arguments and counterarguments that can be constructed from a knowledgebase).

The four approaches described in this issue have many things in common but also differ at several points. For example:

- Two approaches (ABA and ASPIC+) are explicitly within Dung's abstract approach to argumentation, one approach (Besnard & Hunter) leaves various choices at this point open, while DeLP deviates from Dung's approach.
- The four approaches are at varying levels of abstraction. ABA and ASPIC+ (almost) fully abstract from the nature of the logical language and the inference rules and can be instantiated in various ways, deductive argumentation is based on general definitions for any monotonic logic, though particular consideration is given to classical logic and its extensions, while DeLP has a logic-programming like language with rules composed of propositional or first-order literals.
- Two approaches explicitly distinguish between two kinds of inference rules (ASPIC+ and DeLP), called strict and defeasible rules, while ABA reduces defeasible rules to strict rules plus assumptions. In contrast, in deductive argumentation the inference rules are the inference rules of the base logic.
- Finally, three approaches (ASPIC+, DeLP, and deductive argumentation) allow for the use of explicit preferences to resolve attacks, while ABA encodes preferences in assumption-premises of rules.

We refer to the individual papers in this issue for motivations of these various design choices.

### Reference

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