On the Instantiation of Knowledge Bases in Abstract Argumentation Frameworks

Adam Wyner\(^1\), Trevor Bench-Capon\(^2\), and Paul Dunne\(^2\)

\(^1\) Department of Computing Science, University of Aberdeen, Aberdeen, United Kingdom
azwyner@abdn.ac.uk

\(^2\) Department of Computer Science, University of Liverpool, Liverpool, United Kingdom
tbc,sq12@liverpool.ac.uk

Abstract. Abstract Argumentation Frameworks (AFs) provide a fruitful basis for exploring issues of defeasible reasoning. Their power largely derives from the abstract nature of the arguments within the framework, where arguments are atomic nodes in an undifferentiated relation of attack. This abstraction conceals different conceptions of argument, and concrete instantiations encounter difficulties as a result of conflating these conceptions. We distinguish three distinct senses of the term. We provide an approach to instantiating AF in which the nodes are restricted to literals and rules, encoding the underlying theory directly. Arguments, in each of the three senses, then emerge from this framework as distinctive structures of nodes and paths. Our framework retains the theoretical and computational benefits of an abstract AF, while keeping notions distinct which are conflated in other approaches to instantiation.

1 Introduction

Abstract Argumentation Frameworks (AFs) ([1,2,3], among others) provide a fruitful basis for exploring issues of defeasible reasoning. Their power largely derives from the abstract nature of the arguments within the framework, where arguments are atomic nodes in an undifferentiated relation of attack; such AFs provide a very clean acceptability semantics, e.g. [5].

While abstract approaches facilitate the study of arguments and the relations between them, it is necessary to instantiate arguments to apply the theory. In instantiated argumentation, arguments are premises and rules from which conclusions are derived. The objective of such instantiated argumentation is to be able to reason about inconsistency of a knowledge base (KB) and derive consistent subsets of the KB. Methods for instantiation have been proposed which combine AFs with Logic Programs [2,6,7,3,8,9]. Such systems generally have three steps as in Figure 1 (from [10]), though for this paper we focus on the formalisation of ASPIC+ [8]. We start with an inconsistent knowledge base (KB) comprised of facts and rules, where the rules typically may include both strict (SI) and defeasible (DI) inference rules. In Step 1, we construct arguments (nodes) and attacks (arcs) from this KB, resulting in an AF; formalisations differ in just how arguments are constructed from the KB and how attacks between arguments are determined.

\(^1\) Corresponding author: Adam Wyner. This paper is a revision of an unpublished paper [4].

Thanks to Federico Cerutti for comments. Errors and misunderstandings rest with the authors.
In Step 2, we evaluate the AF according to a variety of semantics, resulting in extensions (sets) of arguments. In Step 3, we extract the conclusions from the arguments, resulting in extensions of conclusions. Thus, from a KB that is initially inconsistent (or derives inconsistency), we can nonetheless identify consistent sets of propositions.

![Diagram](image)

**Fig. 1.** Three Steps of Argumentation

While such an approach to instantiated argumentation is attractive, it is not without issues. We discuss these briefly by way of motivation, then develop them over the course of the paper. Arguments in ASPIC+ are constructed from the KB as premises and a rule from which a conclusion is inferred; they may be *compounds* of strict and defeasible subarguments [8]. Thus, many arguments with some of the same elements of the KB may be constructed. An argument may attack a subargument of another argument. Successful attacks (*defeat*) are defined relative to a preference ordering amongst the arguments and used to determine AF extensions. In these respects, ASPIC+ differs from [1], where arguments are atomic, there is a uniform attack relation between arguments, and a preference ordering plays no role in determining successful attack. As well, the use of subarguments and attacks between arguments and subarguments gives rise to some *descriptive* unclarity in the commonly uses “senses” of the term “argument” [11]. More essentially, ASPIC+ must ensure that over the course of the three steps, the *rationality postulates* of direct consistency, closure, and indirect consistency of [3] are satisfied. For ASPIC+ to satisfy the rationality postulates, auxiliary definitions are required and only *restricted rebut* is available [8], though this seems limited [10].

Stepping back from the particulars of ASPIC+, there is a general question of whether all three steps are required to attain the goal of extensions of conclusions; after all, Step 1 “packs” a portion of the KB into arguments that have to be “unpacked” in Step 3. In this way, reasoning with respect to the KB is handled indirectly, with arguments standing as intermediaries. Finally, we cannot reason with *partial information* in KBs, where premises of a rule are missing, for no inference can be drawn, so no argument can be constructed.

In this paper, we provide a novel, two step approach to *instantiating the arguments of an AF* (see Figure 2), where arguments AF are atomic, there are no attacks on subarguments, and preferences are not used. It intuitively satisfies the rationality postulates *without restricted rebut* while addressing a key, problematic example. The AF “wears the logic on its sleeve”: the KB, mainly classical logic with strict and defeasible *modus ponens* to use the rules along with the principles of *ex falso quodlibet* and *tertium non datum*, is directly constructed as an AF with literals and rules as the nodes of the AF, i.e. the arguments of the AF, with arcs, i.e. the attacks of the AF, specified between them. Once given the AF so constructed, evaluation proceeds as usual, though the extensions correlate with *models* of consistent subsets of the KB. We show how we can represent