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.[3] 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

[3] 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 just how arguments are constructed from the KB and how attacks between arguments are determined. 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.

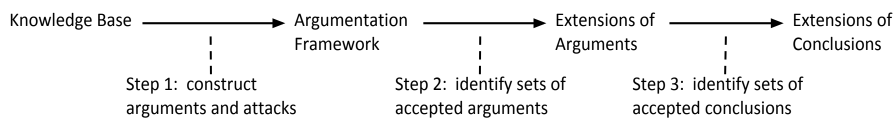

Fig. 1. Three Steps of Argumentation

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 sleave": 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 and reason with partial, incomplete, and inconsistent KBs. Our approach addresses a benchmark example of the ASPIC approach. In addition, in our approach, the various descriptive senses of *argument* such as found in ASPIC+ and elsewhere emerge from the framework as distinct structures in an AF; keeping them distinct avoids the confusions that can arise when these different senses are conflated. Our approach retains the appeal of AFs, evaluates the AF with the well understood semantics, allows reasoning with respect to knowledge bases, retains the appropriate level of abstraction of the nodes of the AF, and reasons with partial KBs.

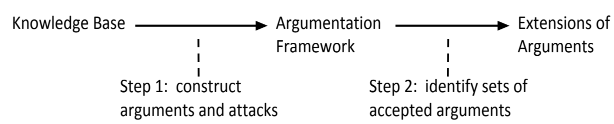

Fig. 2. Two Steps of Argumentation

The structure of the paper is as follows. In Section 2 we outline AFs [1] and characterise the types of knowledge base we are working with. We then show how a knowledge base is represented in a derived AF in Section 3. We illustrate the approach with basic examples of the definitions, a simple example of a combination of strict and defeasible rules, a partial KB, and the relationship of extensions to classical logic models. In Section 4, we discuss the approach to KB instantiation of [3,8] along with a key example and the problems it raises. We show how our approach addresses the problems of the example. The different senses of *argument* are then characterised in terms of particular structures within the AF as presented in Section 5. We end in Section 6 with some concluding remarks and future work.

2 Argumentation Frameworks

An *Argumentation Framework* AF is defined as follows [1].

Definition 1. *An argumentation framework* AF *is a pair* $\langle \mathcal{L}^A, \mathcal{R}^A \rangle$*, where* $\mathcal{L}^A$ *is a finite set of* arguments, $\{p_1, p_2, \ldots, p_n\}$ *and* $\mathcal{R}^A$ *is an* attack *relation between elements of* $\mathcal{L}^A$*. For* $\langle p_i, p_j \rangle \in \mathcal{R}^A$ *we say the argument* p_i *attacks argument* p_j*. We assume that no object attacks itself.*

The relevant auxiliary definitions are as follows, where S is a subset of $\mathcal{L}^A$:

Definition 2. *We say that* $p \in \mathcal{L}^A$ *is* acceptable with respect to S *if for every* $q \in \mathcal{L}^A$ *that attacks* p *there is some* $r \in S$ *that attacks* q*. A subset,* S*, is* conflict-free *if no argument in* S *is attacked by any other argument in* S*. A conflict-free set* S *is* admissible *if every* $p \in S$ *is acceptable to* S*. A* preferred extension *is a maximal (w.r.t.*

$\subseteq$) *admissible set. The argument* $p \in \mathcal{L}^A$ *is* credulously accepted *if it is in at least one preferred extension, and* sceptically accepted *if it is in* every *preferred extension.*

There are a variety of other semantics, e.g. *grounded*, *stable*, and others, but considering preferred extensions serves our purposes in this paper.

As we clarify the notion of *argument* itself and do not want to introduce presumptions about them, we sometimes prefer to refer to arguments as *objects* or graph-theoretic *nodes* (denoted by $\mathcal{L}^A$) and their attack relations as *arcs* (denoted by $\mathcal{R}^A$). Context makes it clear what is being referred to.

3 Representing a Theory as an AF

The approach has two basic parts (the presentation is a revision of [4]). In the first part, we represent a Theory Base $\mathcal{T}$, which represents the KB. Then, we construct an AF from the KB, following Step 1 of Figure 2, where the nodes of an AF are labeled with respect to the literals and inference rules of the Theory Base, while the attack relation is partitioned with respect to the nodes. In the second part, we impose conditions on the assertion of literals with respect to the AF. Following the theoretical presentation, we provide basic examples, carrying out Step 2 of Figure 2 to evaluate an AF according to Definitions 1 and 2.

3.1 Theory Base $\mathcal{T}$

Definition 3. *A* Theory Base, $\mathcal{T}$, *comprises a pair* $(\mathcal{L}, \mathcal{R})$ *in which*

$$\mathcal{L} = \{x_1, \ldots, x_n\} \cup \{\neg x_1, \ldots, \neg x_n\}$$

is a set of literals over a set of propositional variables $\{x_1, \ldots, x_n\}$. *We use* y_i *to denote an arbitrary literal from* $\{x_i, \neg x_i\}$.

We have a set of proper names *of rules* $\{r_1, r_2, \ldots, r_n\}$. *Rules are either strict (*$r \in \mathcal{R}_{str}$*) or defeasible (*$r \in \mathcal{R}_{dfs}$*), and* $\mathcal{R}_{str} \cap \mathcal{R}_{dfs} = \emptyset$. $\mathcal{R} = \mathcal{R}_{str} \cup \mathcal{R}_{dfs}$ *where*

$$\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$$

in which $r \in \mathcal{R}$ *has a body, bd(r)* $\subseteq \mathcal{L}$*, and a head, hd(r)* $\in \mathcal{L}$*.*

We refer to the literals in bd(r) as *premises* and the literal in hd(r) as the *claim*.

For easy reference to the "content" of the rule, we assume each rule has an associated *definite description* as follows. For $r \in \mathcal{R}_{str}$, the definite description of r has the form $r : bd(r) \rightarrow hd(r)$, where $hd(r) \in \mathcal{L}$ and $bd(r) \subseteq \mathcal{L}$. Similarly, the definite description for $r \in \mathcal{R}_{dfs}$, has the form $r : bd(r) \Rightarrow hd(r)$. Where a rule has an empty body, bd(r) = $\emptyset$, we have $r :\rightarrow hd(r)$ or $r :\Rightarrow hd(r)$, which are *strict* and *defeasible* assertions, respectively. To refer distinctly to the set of rules with non-empty bodies and those with empty bodies (*assertions*), we have $\mathcal{R}$ = TRules $\cup$ ARules, where TRules = $\{r \mid r \in \mathcal{R} \wedge bd(r) \neq \emptyset\}$ and ARules = $\{r \mid r \in \mathcal{R} \wedge bd(r) = \emptyset\}$.

We constrain a Theory Base, which we refer to as a *Well-formed Theory*.

Definition 4. *A* Well-formed Theory*, $\mathcal{W}$, is a Theory Base, $\mathcal{T}$, abiding Constraints 1-4.*

First, the relationship between literals of strict and defeasible rules is constrained:

Constraint 1 *For* Theory Base $(\mathcal{L}, \mathcal{R})$*, $\forall r \in \mathcal{R}_{str}$, there is no rule, r' $\in \mathcal{R}_{dfs}$ with hd(r) = hd(r') and bd(r) $\subseteq$ bd(r').*

Furthermore, no literal and its negation can both be strictly asserted.

Constraint 2 *For* Theory Base $(\mathcal{L}, \mathcal{R})$*, if r $\in \mathcal{R}$, where r :$\rightarrow$ hd(r), then r' $\notin \mathcal{R}$, where r' :$\rightarrow$ $\neg$hd(r).*

In addition, every literal appears in some rule.

Constraint 3 *For* Theory Base $(\mathcal{L}, \mathcal{R})$*, if y $\in \mathcal{L}$, then $\exists$ r $\in \mathcal{R}$, y $\in$ bd(r) $\vee$ y = hd(r).*

Finally, every rule has a claim.

Constraint 4 *For* Theory Base $(\mathcal{L}, \mathcal{R})$*, if r $\in \mathcal{R}$, then $\exists$ y $\in \mathcal{L}$, y = hd(r).*

Semantically, a rule $r \in \mathcal{R}_{str}$ represents the notion that hd(r) holds if *all* of the literals in bd(r) simultaneously hold; with respect to the rule, we say the bd(r) strictly implies the hd(r). We assume standard notions of *truth* and *falsity* of literals along with the truth-tables of Propositional Logic for material implication which are models under which the rule is *true* or *false*. Semantically, a rule $r \in \mathcal{R}_{dfs}$ represents the notion that hd(r) "usually" holds if *all* of the literals in bd(r) simultaneously hold, but there are circumstances where ¬hd(r) holds though *all* of the literals in bd(r) simultaneously hold. With respect to the rule, we say the bd(r) defeasibly implies the hd(r).

While the clauses are similar to the *Horn Clauses* of logic programming, the head literal can be in a positive or negative form. We only have classical negation, not negation as failure; we do not allow iterated negation. The rationale for this choice of clauses is that it naturally supports our analysis of the senses of *argument*.

3.2 Deriving an AF from a Theory Base

A core element of our approach is the concept of the AF derived from a Theory Base. The AF uses a set of labels for the nodes in the graph: $\{\mathsf{x}_1,\ldots,\mathsf{x}_n\} \cup \{\neg\mathsf{x}_1,\ldots,\neg\mathsf{x}_n\} \cup \{\mathsf{r}_1,\ldots,\mathsf{r}_n\}$ (or for clarity, the definite description of the rule name). Thus, we can see how elements of a Theory Base, $\mathcal{T}$, correspond to but are distinct from elements of the derived AF, indexing the AF to the $\mathcal{T}$.

Definition 5. *Let $\mathcal{T} = (\mathcal{L}, \mathcal{R})$ be a Theory Base with*

$$\begin{aligned}\mathcal{L} &= \{x_1, \ldots, x_n\} \cup \{\neg x_1, \ldots, \neg x_n\}\\ \mathcal{R} &= \mathcal{R}_{str} \cup \mathcal{R}_{dfs}\end{aligned}$$

The derived framework *from $\mathcal{T}$, is the* AF, $\langle \mathcal{L}^A_{\mathcal{T}}, \mathcal{R}^A_{\mathcal{T}} \rangle$ *in which,*

$$\begin{aligned}\mathcal{L}^A_{\mathcal{T}} = \ & \{\ \mathsf{x},\ \neg\mathsf{x}\ : x, \neg x\ \in \mathcal{L}\}\\ & \cup \{\ \textsf{\textit{r : bd(r)}} \rightarrow \textsf{\textit{hd(r)}}\ :\ r\ \in \mathcal{R}_{str}\}\\ & \cup \{\ \textsf{\textit{r : bd(r)}} \Rightarrow \textsf{\textit{hd(r)}}\ :\ r\ \in \mathcal{R}_{dfs}\}\end{aligned}$$

Furthermore,

$$\begin{aligned}&\forall \mathsf{x} \in \mathcal{L}^A_{\mathcal{T}}, x\ \in \mathcal{L},\ and\\ &\forall \mathsf{r} \in \mathcal{L}^A_{\mathcal{T}}, r\ \in \mathcal{R}\end{aligned}$$

In an AF, the nodes have no internal content.

The attack set $\mathcal{R}^A_{\mathcal{T}}$ comprises three disjoint sets which describe: attacks by nodes labeled with names for literals on other nodes labeled with names for literals; attacks by nodes labeled with names for literals on nodes labeled with names for rules; and attacks by nodes labeled with names for rules on nodes labeled with names for literals. We recall that $y_i \in \{x_i, \neg x_i\}$ so that $\neg y_i$ is the complementary literal to y_i.

Definition 6. *In the* AF $\langle \mathcal{L}^A_{\mathcal{T}}, \mathcal{R}^A_{\mathcal{T}} \rangle$, $\mathcal{R}^A_{\mathcal{T}} = \mathcal{R}^A_{ll} \cup \mathcal{R}^A_{lr} \cup \mathcal{R}^A_{rl}$ *where:*

$$\begin{aligned}\mathcal{R}^A_{ll} &= \{\langle \mathsf{y}_i, \neg\mathsf{y}_i \rangle,\ \langle \neg\mathsf{y}_i, \mathsf{y}_i \rangle\ :\ 1 \leq i \leq n\\ &\quad\ and\ \mathsf{y}_i, \neg\mathsf{y}_i \in \mathcal{L}^A_{\mathcal{T}}\}\\ \mathcal{R}^A_{lr} &= \{\langle \neg\mathsf{y}_i, \mathsf{r}_j \rangle\ :\ y_i \in bd(r_j)\ and\ \neg\mathsf{y}_i, \mathsf{r}_j \in \mathcal{L}^A_{\mathcal{T}}\}\\ &\quad \cup \{\langle \neg\mathsf{y}_i, \mathsf{r}_j \rangle\ :\ r_j \in \mathcal{R}_{dfs}\ and\ hd(r_j) = y_i\\ &\quad\ and\ \neg\mathsf{y}_i \in \mathcal{L}^A_{\mathcal{T}}\}\\ \mathcal{R}^A_{rl} &= \{\langle \mathsf{r}_j, \neg\mathsf{y}_i \rangle\ :\ hd(r_j) = y_i\ and\ \neg\mathsf{y}_i, \mathsf{r}_j \in \mathcal{L}^A_{\mathcal{T}}\}\end{aligned}$$

The following hold for an AF derived from a $\mathcal{T}$:

1. Each literal y in $\mathcal{L}$ of Theory Base $\mathcal{T}$ corresponds to a node labeled y in $\mathcal{L}^A$ of the derived AF; $\mathcal{L}^A$ of the derived AF contains, in addition, the node labeled $\neg\mathsf{y}$. Nodes labeled for literals of opposite polarity are mutually attacking.
2. Each rule in r in $\mathcal{R}$ of a Theory Base $\mathcal{T}$ corresponds one-to-one to a node label r in $\mathcal{L}^A$ of the derived AF. Whereas a rule in $\mathcal{R}$ is true (or false) in the Theory Base, in the derived AF we say it *has been applied* relative to the admissible set where it appears and otherwise *has not been applied*. In the AF, a rule node is attacked by the nodes which correspond to the negation of the body literals and, in addition, attacks the node which corresponds to the negation of the head literal.
3. For *strict* rules, if a node which corresponds to the negation of a body literal of a rule is in an admissible set, we say the rule node has not been applied relative to that set. In this case, the node which corresponds to the head literal is only credulously admissible. If all the nodes which correspond to the body literals are in an admissible set, then the rule node has been applied and the node which corresponds to head literal is admissible in that set.

4. For *defeasible* rules, if a node which corresponds to the negation of a body literal or if the node which corresponds to the negation of the head literal of the rule is in an admissible set, we say the rule node has not been applied relative to that set. In both instances, the node corresponding to the literal attacks the rule node. Even if all nodes which correspond to the body literals of a rule are in an admissible set, the rule node or the node which corresponds to the head literal may not be in that set, for they can be defeated.

We evaluate the derived AFs only following the definitions of extensions relative to the standard AF $\langle \mathcal{L}^A_{\mathcal{T}}, \mathcal{R}^A_{\mathcal{T}} \rangle$; that is, while the partitions of nodes or arcs are important for deriving the AF from $\mathcal{T}$, they are ignored for the purposes of the standard AF evaluation, so that we have a standard abstract framework. Thus, the fundamental semantics of abstract AFs are maintained.

For our purposes and relative to our classical logic context, the set of extensions provided by Dungian AFs must be *filtered*. In our approach, AFs are derived from a Theory Base, and the resulting extensions are *not* homogeneous, for they may contain both literals and rules. More importantly, we must ensure that the extensions also serve to satisfy classical logic properties such as *closure* under strict implication. With these points in mind, we have the following.

Constraint 5 *Consider: a, an admissible set of the derived* AF $\langle \mathcal{L}^A_{\mathcal{T}}, \mathcal{R}^A_{\mathcal{T}} \rangle$*; $\mathcal{A}$, the set of admissible sets a; and $\mathcal{R}_{str} \subseteq \mathcal{L}^A_{\mathcal{T}}$. For every $r \in \mathcal{R}_{str}$ and every $a \in \mathcal{A}$, if $r \in a$ and every bd(r) $\in a$, then hd(r) $\in a$.*

Definition 7. *An admissible set of the derived* AF $\langle \mathcal{L}^A_{\mathcal{T}}, \mathcal{R}^A_{\mathcal{T}} \rangle$ *is a* Well-formed Admissible Set *(WFAS) iff it satisfies Constraint 5.*

The implication is that relative to WFASs, the hd(r) of a rule r is *sceptically acceptable* relative to the derived AF. On the other hand, for $r' \in \mathcal{R}_{dfs}$, hd(r') is *credulously acceptable* relative to the derived AF. We emphasise that we change nothing about Dungian AFs or evaluations, but we do provide a justification to select amongst the resulting extensions. These points are illustrated with respect to Figure 6.

To this point, we have Theory Bases, corresponding derived AFs, and a constraint on extensions. Fundamental observations of our approach are:

Observation 1 *For the literals and the rules which are* true *of every model for the Theory Base $\mathcal{T}$, the corresponding nodes of the WFAS extensions of the derived* AF *are sceptically acceptable, otherwise they are credulously acceptable.*

Observation 2 *For the literals and the rules which are* false *in any model of a Theory Base $\mathcal{T}$, the corresponding nodes of the WFAS extensions of the derived* AF $\langle \mathcal{L}^A_{\mathcal{T}}, \mathcal{R}^A_{\mathcal{T}} \rangle$ *are not an element of any admissible set.*

Both of these follow by the evaluation of a derived framework $\langle \mathcal{L}^A_{\mathcal{T}}, \mathcal{R}^A_{\mathcal{T}} \rangle$ relative to a $\mathcal{T}$. Thus, the derived AF is *information-preserving* with respect to the Theory Base. The derived AF is an instantiation of the corresponding Theory Base, and the preferred extensions of the AF correspond to models of the Theory Base.

3.3 Examples of the Definitions

We now give some examples of the basic definitions, discuss defeasibility, provide a simple combination of strict and defeasible rules, illustrate reasoning with an assertion in a partial KB, and comment on the connection between the extensions and the classical models. In Section 4, we give a more complex, problematic example from the literature is used to illustrate the advantages of this approach over an ASPIC-type approach. First, we provide a Theory Base $\mathcal{T}_1$ with just one strict rule, the derived AF, a graphic representation of the derived AF, and then the preferred extensions. Since it is always clear in context where we have literals and rules (in a Theory Base) and where we have labels (in an AF), we use one typographic form without confusion.

Example 1. Let $\mathcal{T}_1$ be the pair with $(\mathcal{L}_1, \mathcal{R}_1)$, where

$$\begin{aligned} \mathcal{L}_1 &= \{x_1, x_2\} \cup \{\neg x_1, \neg x_2\} \\ \mathcal{R}_1 &= \{r_1\},\ where\ r_1\ has\ rule\ name\ r_1 : x_1 \rightarrow x_2 \end{aligned}$$

The *derived framework* from $\mathcal{T}_1$ is $\langle \mathcal{L}^A_{\mathcal{T}_1}, \mathcal{R}^A_{\mathcal{T}_1} \rangle$ in which,

$$\mathcal{L}^A_{\mathcal{T}_1} = \{x_1, x_2\} \cup \{\neg x_1, \neg x_2\} \cup \{r_1\}$$

and in which $\mathcal{R}^A_{\mathcal{T}_1}$ comprises the union of three disjoint sets:

$$\begin{aligned} \mathcal{R}^A_{ll} &= \{\langle x_1, \neg x_1\rangle, \langle \neg x_1, x_1\rangle, \langle x_2, \neg x_2\rangle, \langle \neg x_2, x_2\rangle\} \\ \mathcal{R}^A_{lr} &= \{\langle \neg x_1, r_1\rangle\} \\ \mathcal{R}^A_{rl} &= \{\langle r_1, \neg x_2\rangle\} \end{aligned}$$

We graphically represent $\langle \mathcal{L}^A_{\mathcal{T}_1}, \mathcal{R}^A_{\mathcal{T}_1} \rangle$ as in Figure 3.

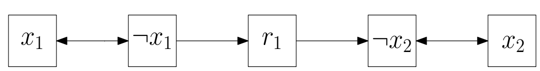

Fig. 3. AF of $x_1 \rightarrow x_2$

In $\langle \mathcal{L}^A_{\mathcal{T}_1}, \mathcal{R}^A_{\mathcal{T}_1} \rangle$, the preferred extensions are:

$$\{x_1, r_1, x_2\}, \{\neg x_1, x_2\}, \{\neg x_1, \neg x_2\}$$

Each of the nodes is *credulously accepted* and none is *sceptically* accepted. The interpretation of the presence of a rule node in a preferred extension is that the rule *has been applied*. Moreover, the rule is not *defeated* in the sense that where the premises hold, the conclusion *must* hold. No admissible set contains both x_1 and $\neg x_2$: if x_1 is in the set, then r_1 is in the set; r_1 attacks $\neg x_2$, leaving x_2 in the set; if $\neg x_2$ is in the set, then r_1 must be attacked; r_1 can only be attacked by $\neg x_1$, which also attacks x_1, leaving $\neg x_1$ in the set.

There are three related points about the extensions. First, we can provide extensions in an AF *with respect to a rule per se*; that is, it is not necessary to provide *asserted* premises along with the rule from which we draw the inferred conclusion. By the same token, we can provide extensions where only some of the premises are asserted, e.g. the KB has partial, incomplete information of what holds. Suppose a Theory Base with only the following two rules: $\rightarrow x_1$ and $r_4{:}x_1, x_3 \rightarrow x_2$. The extensions are:

$$\{x_1, x_3, r_1, x_2\}, \{x_1, \neg x_3, x_2\}, \{x_1, \neg x_3, \neg x_2\}$$

In this respect, our approach differs markedly from approaches to instantiated AFs that rely on KBs where inferences are essential to the construction of well-formed arguments. Third, we see that the preferred extensions with respect to an AF correlate with the models of the Theory Base; in this respect, the AF and Step 2 of Figure 2 can be viewed as a means to build models for the Theory Base. These three points apply to strict and defeasible rules alike.

The following is an example of a *defeasible* rule.

Example 2. Let $\mathcal{T}_2$ be the pair with ($\mathcal{L}_2$, $\mathcal{R}_2$), where

$$\begin{aligned} \mathcal{L}_2 &= \{x_1, x_2\} \cup \{\neg x_1, \neg x_2\} \\ \mathcal{R}_2 &= \{r_2\},\ where\ r_2\ has\ rule\ name\ r_2 : x_1 \Rightarrow x_2 \end{aligned}$$

We graphically represent the derived AF $\langle \mathcal{L}^A_{\mathcal{T}_2}, \mathcal{R}^A_{\mathcal{T}_2} \rangle$ as: In $\langle \mathcal{L}^A_{\mathcal{T}_2}, \mathcal{R}^A_{\mathcal{T}_2} \rangle$, the preferred

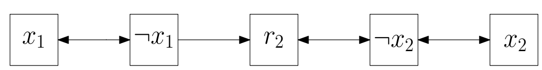

Fig. 4. AF of $x_1 \Rightarrow x_2$

extensions are as follows, where we see that each of the nodes is *credulously accepted* and none is *sceptically* accepted.

$$\{x_1, r_2, x_2\}, \{\neg x_1, x_2\}, \{\neg x_1, \neg x_2\}, \{x_1, \neg x_2\}$$

The first three preferred extensions are similar to SI. In the last extension, $\neg x_2$ itself attacks the rule node r_2; consequently, either x_1 or $\neg x_1$ are in a preferred extension along with $\neg x_2$. This contrasts with the preferred extension of a derived AF with just a SI. While defeasible implication might be construed as the trivial logical tautology $[x_1 \rightarrow [x_2 \vee \neg x_2]]$, here we see a key difference, which highlights the utility of some semantic content to the extensions. To make use of a defeasible rule, one must provide the means to *choose between extensions*, for example, by selecting the extension which maximises the number of applicable defeasible rules, or which uses some notion of priority or entrenchment on the rules. Different ways of making this choice give rise to different varieties of non-monotonic logic [12,13]). Circumscription [14] could be used by including additional designated nodes such as $ab(r_1)$ which attack the rule r_1

and attack and are attacked by $notab(r_1)$. We then choose the extension containing the most $notab(r_1)$ nodes. Using DeLP defeaters [6], we can specify circumstances where the rule is not applied.

In our third example, we show the interaction of defeasible and strict rules, which was the root of several of the problems identified in [3].

Example 3. Suppose $\mathcal{T}_3$ with rules r$_2$: x$_1 \Rightarrow$ x$_2$ and r$_3$: x$_2 \rightarrow$ x$_3$ which has derived AF $\langle \mathcal{L}^A_{\mathcal{T}_3}, \mathcal{R}^A_{\mathcal{T}_3} \rangle$ graphically represented as in Figure 5.

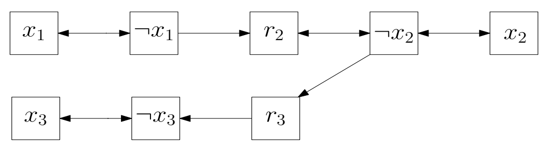

Fig. 5. AF derived from $\mathcal{T}$ with $x_1 \Rightarrow x_2$ and $x_2 \rightarrow x_3$

AF $\langle \mathcal{L}^A_{\mathcal{T}_3}, \mathcal{R}^A_{\mathcal{T}_3} \rangle$ has the following six preferred extensions:

1.$\{x_1, r_2, x_2, r_3, x_3\}$	4.$\{\neg x_1, x_2, r_3, x_3\}$
2.$\{x_1, \neg x_2, x_3\}$	5.$\{\neg x_1, \neg x_2, x_3\}$
3.$\{x_1, \neg x_2, \neg x_3\}$	6.$\{\neg x_1, \neg x_2, \neg x_3\}$

Given a strict assertion that x_1, we would normally choose the preferred extension (1) from among (1)-(3), maximising the number of defeasible rules. Thus, normally, we say that x_1 implies x_3. However, we are not obliged to make this choice. In particular, if $\neg x_2$ is strictly asserted, r_2 and r_3 are inapplicable, and x_3 is credulously acceptable ((2) and (3)); thus, in this AF, a strict assertion of x_1 does not imply that x_3 necessarily holds as well. Where the claim of a defeasible rule is a premise of a strict rule (x_2), we cannot use the defeasibly inferred claim to draw strict inferences about the claim of the strict rule (x_3). Similarly, the defeasible rule is inapplicable where either the claim of the rule ($\neg x_2$) is false ((2), (3), (5), and (6)) or the claim of the strict rule ($\neg x_3$) is false ((3) and (6)). Whereas in e.g. [12], the defeasible rule is inapplicable only where the claim of the defeasible rule itself is asserted to be false, here the falsity of any consequences of that claim, however remote, will also block the application of the rule.

4 Comparison to ASPIC with a Base-case Example

In this section, we briefly review the key components of the benchmark argument instantiation method of [3,8], compare it to our proposal, provide one of the key examples which showed a flaw in the instantiation method as well as motivated the *Rationality Postulates*. We then show how such problems do not arise in our approach.

In constructing arguments, several functions are introduced: `Prem` is the set of premisese of the argument, `Conc` returns the last conclusion of an argument, `Sub` returns all the subarguments of an argument, `DefRules` returns all the defeasible rules used in an argument, and `TopRule` returns the last inference rule used in the argument. Theory Bases $\mathcal{T}$ are comprised of strict and defeasible implications. Arguments have a deductive form and are constructed recursively from the rules of the Theory Base. To distinguish strict or defeasible rules from the deductive form of arguments, we use *short* arrows, $\rightarrow$ and $\Rightarrow$, for the former and *long* arrows, $\longrightarrow$ and $\Longrightarrow$ for the latter. For brevity, we only provide the clauses for the construction of strict arguments as the clauses for the construction of defeasible arguments are analogous (including among the DefRules the TopRule(A) that is defeasible) [8].

Definition 8. *(Argument) Suppose a* Theory Base*, $\mathcal{T}$, with strict and defeasible rules. An argument* A *is:*
$A_1, \ldots, A_n \longrightarrow \psi$ if $A_1, \ldots, A_n$, with $n \geq 0$, are arguments such that there exists a strict rule `Conc`(A_1), ..., `Conc`(A_n) $\rightarrow \psi$.
`Prem`(A) = `Prem`(A_1) $\cup \ldots \cup$ `Prem`(A_n),
`Conc`(A) = ψ,
`Sub`(A) = `Sub`(A_1) $\cup \ldots \cup$ `Sub`(A_n) $\cup \{A\}$,
`DefRules`(A) = `DefRules`(A_1) $\cup \ldots \cup$ `DefRules`(A_n)
`TopRule`(A) = `Conc`(A_1), ..., `Conc`(A_n) $\rightarrow \psi$

Consider a Theory Base with strict and defeasible rules from which we construct arguments according to this definition (see Example 5 [3]).

Example 4. Let $\mathcal{T}_4$ be a Theory Base with the following rules:
r_{21}: $\rightarrow x_1$; r_{22}: $\rightarrow x_2$; r_{23}: $\rightarrow x_3$; r_{24}: $x_4, x_5 \rightarrow \neg x_3$; r_{25}: $x_1 \Rightarrow x_4$; r_{26}: $x_2 \Rightarrow x_5$.
We construct the following arguments:
A_1: $[[\rightarrow x_1] \Rightarrow x_4]$; A_2: $[[\rightarrow x_2] \Rightarrow x_5]$; A_3: $[\rightarrow x_3]$;
A_4: $[\rightarrow x_1]$; A_5: $[\rightarrow x_2]$;
A_6: $[[\rightarrow x_1] \Rightarrow x_4], [[\rightarrow x_2] \Rightarrow x_5] \rightarrow \neg x_3$.

We see clearly that arguments can have subarguments: A_6 has a subargument A_1, and A_1 has a subargument A_4.

Several additional elements are needed to define *justified conclusions*. An argument is strict if it has no defeasible subargument, otherwise it is defeasible (non-strict). An argument A_i rebuts an argument A_j where the conclusion of some subargument of A_i is the negation of the conclusion of some non-strict subargument of A_j; rebuttal is one way an argument defeats another argument. Admissible argument orderings specify that a strict argument (containing premises that are axioms and rules that are strict) can defeat a defeasible argument, but not vice versa. Moreover, one argument can defeat another argument with respect to subarguments; in effect, defeat of a part is inherited as defeat of a whole. With respect to our example, the undefeated arguments are A_1, A_2, A_3, A_4, and A_5. A_3, which is a strict argument, defeats A_6 but not vice versa since A_6 is a non-strict argument in virtue of having a defeasible subargument. Given the arguments and defeat relation between them, we can provide an AF and the different extensions. The `Output` of an AF, understood as the *justified conclusions* of the AF, is given as the sceptically accepted conclusions of the arguments of the AF.

With respect to the example, [3] claim that the justified conclusions are x_1, x_2, x_3, x_4, and x_5 since these are all conclusions of arguments which are not attacked (it appears not to be an example analysed in [8]). However, $\neg x_3$ is *not* a justified conclusion, even though it is the conclusion of a strict rule in which all the premises are justified conclusions. This is so since the argument A_6 of which $\neg x_3$ is the conclusion is defeated by but does not defeat A_3 because A_6 has a *subargument* which is a non-strict argument (namely A_1 or A_2), so making A_6 a non-strict argument, while A_3 is a strict argument. Yet, given the antecedents of the strict rule are justified conclusions, it would seem intuitive that the claim of a strict rule should also be a justified conclusion. This, they claim, shows that justified conclusions are not closed under strict rules or could even be inconsistent.

In our view, these notions of argument and defeat are problematic departures from [1], which has no notion of subargument or of defeat in terms of subarguments. In addition, they give rise to the problems with justified conclusions: what is a strict rule in the Theory Base can appear in the AF as a non-strict argument in virtue of subarguments; what cannot be false in the Theory Base without contradiction is defeated in the AF; thus, what "ought" to have been a justified conclusion is not. In addition, the notion of justified conclusion leads to some confusion: on the one hand, it only holds for sceptically accepted arguments, which presumably implies that the propositions which constitute them are sceptically accepted; on the other hand, there is no reason to expect that $\neg x_3$ is sceptically accepted, given that it only follows from defeasible antecedents. Clearly the anomaly arises because of the way that arguments can have defeasible subarguments, that the defeat of the whole can be determined by the defeat of a part, and that justified conclusions depend on these notions.

In our approach, the results are straightforward and without anomaly; we do not make use of arguments with subarguments, inheritance of defeasiblity, or problematic notions of justified conclusions. We consider a key example from [3] as the two other problematic examples cited in [3] follow suit. The Theory Base of Example 4 appears as in Figure 6, for which all the preferred extensions for the AF are given. For clarity and discussion, we include undefeated strict and defeasible rules.

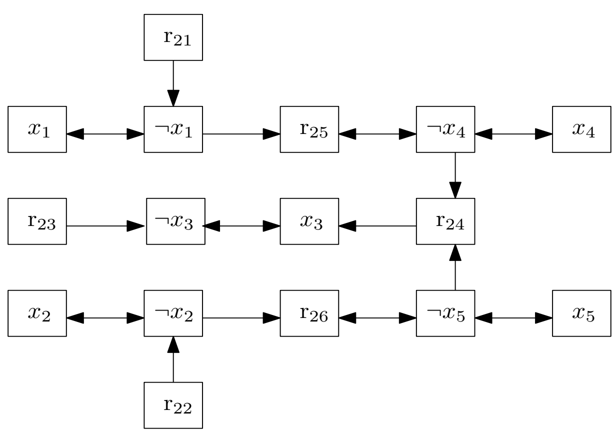

Fig. 6. Graph of Problem Example

$1.\{x_1, r_{21}, r_{25}, x_2, r_{22}, x_3, x_4, r_{25}, \neg x_5\}$
$2.\{x_1, r_{21}, r_{23}, x_2, r_{22}, x_3, \neg x_4, x_5, r_{26}\}$
$3.\{x_1, r_{21}, r_{23}, x_2, r_{22}, x_3, \neg x_4, \neg x_5\}$
$4.\{x_1, r_{21}, r_{23}, r_{25}, x_2, r_{22}, r_{24}, r_{26}, x_4, x_5\}$

Notice that extensions (1)-(3) are unproblematic with respect to *consistency* and *closure*. They also satisfy Definition 7, so are the *relevant* extensions to consider. In contrast, extension (4) is problematic in an argumentation theory *without* Definition 7 since the conclusions of strict rules are missing, thus violating *closure*. Yet, (4) does not satisfy Constraint 5: the premises and rule nodes of strict rules are present, but the conclusions are not. With respect to those extensions that satisfy Definition 7, x_1, x_2, x_3 are all sceptically accepted, while x_4 and x_5 are credulously accepted. $\neg x_3$ is not credulously accepted given that x_3 is strictly asserted. Note that every literal which is strictly asserted is sceptically acceptable. Therefore, the rule node r_{24} must be defeated where one or both of $\neg x_4$ and $\neg x_5$ hold. There is, in our view, no reason to expect $\neg x_3$ to hold in any extension since we have no preferred extension in which both x_4 and x_5 are justified conclusions. Given admissible sets, we satisfy the *consistency* rationality postulate; *closure*, which is relevant only of strict rules where all the body literals hold, is not relevant to this problem. Moreover, we can provide machinery to meet Definition 7 in that we can examine the extensions relative to the rules of the AF to determine if Constraint 5 is satisfied. The analysis also corresponds well with model-building for classical logic.

We have considered a widely adopted approach to instantiating Theory Bases in AFs [8] along with the problems that arise. There are other approaches to instantiating a KB in an AF that may avoid problems with the Rationality Postulates such as Assumption-based [2] or Logic-based [7] argumentation. We leave further comparison and contrast to future work. However, these approaches, like the ASPIC approach, follow the three step structure of Figure 1.

5 Three Senses of Argument

In this section, we discuss the auxiliary point about the various conflated *senses* of the term *argument* as found in the literature. We show how these senses can be formally articulated in our framework as distinct structures [4] . The term *argument* is ambiguous [11]. It can mean the reasons for a claim given in one step (an *Argument*); or it can mean a train of reasoning leading towards a claim (a *Case*), that is, a set of linked *Arguments*; or it can be taken as reasons for and against a claim (a *Debate*), that is a *Case* for the claim and a *Case* against the claim. An additional structure is where the intermediate claims of the *Debate* are also points of dispute, but we will not consider this further here. In the following, we formally define these three senses of *argument* as structures in the argumentation framework, starting with *Arguments*, then providing *Cases*, and finally *Debates*. We provide a graphic, examples, and then definitions for the three different kinds of attack: *Rebuttal*, *Undercut*, and *Premise Defeat*.

We provide a recursive, pointwise definition of a graph which is constructed relative to an AF. Since the sets are constructed relative to an AF, we can infer the attack relations which hold among them. The different senses of *argument* are defined as subgraphs.

Definition 9. *Suppose there is a derived* AF = $\langle \mathcal{L}^A, \mathcal{R}^A \rangle$*, where* y *and* z *are arbitrary literals from* $\mathcal{L}^A$ *and r and r' are arbitrary rules from* $\mathcal{L}^A$*.* $\mathcal{F}$ *abbreviates* $\{r : r \; added \; in \; \rho_{2k-1}\}$.

$$\begin{aligned}
\rho_0(y) &= \{y, \neg y\} \\
\rho_1(y) &= \rho_0(y) \cup \bigcup_{\{r:hd(r)=y\}} \{r\} \\
\rho_{2k}(y) &= \rho_{2k-1}(y) \cup \bigcup_{\{r \in \mathcal{F}\}} \{z, \neg z : z \in bd(r)\} \\
\rho_{2k+1}(y) &= \rho_{2k}(y) \cup \bigcup_{\{r \in \mathcal{F}\}} \{r' : z \in hd(r') \cap bd(r)\} \\
\rho_{2k+2}(y) &= \rho_k(y)
\end{aligned}$$

$\rho_0(y)$ provides the basis for the construction, which are nodes labeled by literals in an AF that attack one another with respect to the node labeled y. At $\rho_1(y)$, we add to the previous set of rules which have y as their head; depending on whether we have a strict or a defeasible rule, the rule node attacks and may be attacked by the literal which is the negation of the head. At $\rho_{2k}(y)$, we add the positive and negative literals relative to the body of the rules; each of the negative literals associated with literals of the body of the rule attacks the rule node. At $\rho_{2k+1}(y)$, we link rules: the literals in the body of a rule added at $\rho_1(y)$ serve as the heads of other rules. At $\rho_{2k+2}(y)$, we have iterated the steps $\rho_1(y)$-$\rho_{2k+1}(y)$ until there is no further change. Constructions for negations of literals are similarly defined.

Supposing a derived AF, Arg_{S1} and Arg_{S2} are subgraphs of that AF. An *Argument for* y, $Arg_{S1}(y)$, is defined at $\rho_{2k}(y)$: it is the nodes and their attacks defined at this step relative to the derived AF. A graph defined as $Arg_{S1}(y)$ can *only have one rule in the set of nodes*, namely a rule of the Theory Base with y as head (other rules with y as head will give rise to distinct arguments for y in sense 1). In $Arg_{S1}(y)$, y is the *claim* of $Arg_{S1}(y)$ and the literals in the body of the rule are the *premises*. A *Case for* y, $Arg_{S2}(y)$, is defined where $\rho_{k+1}(y) = \rho_k(y)$. $Arg_{S2}(y)$ is comprised of $Arg_{S1}(y)$ along with graphs of form Arg_{S1} for the literals that are bodies of every rule constructed relative to $Arg_{S1}(y)$. In other words, a Case links together all those graphs of Arguments for a particular y where the claim of one rule is the premise of another rule.

Definition 10. *Suppose an* AF *derived from Theory Base* $\mathcal{T}$, $\langle \mathcal{L}^A_\mathcal{T}, \mathcal{R}^A_\mathcal{T} \rangle$*. We define* Arg_{S1}-Arg_{S2} *as subgraphs of a derived* AF*:*

$$\begin{aligned}
&An\ Argument\ for\ y\ is\ Arg_{S1}(y) \;=\; \langle \mathcal{L}^A_{S_1y},\ \mathcal{R}^A_{S_1y} \rangle, \\
&where\ \mathcal{L}^A_{S_1y} \subseteq \mathcal{L}^A_\mathcal{T}\ and\ \mathcal{R}^A_{S_1y} \subseteq \mathcal{R}^A_\mathcal{T}, \\
&\forall r,\ r' \in \mathcal{L}^A_{S_1y}\ r \;=\; r',\ is\ a\ subgraph\ at\ \rho_{2k}(y).
\end{aligned}$$

$$\begin{aligned}
&A\ Case\ for\ y\ is\ Arg_{S2}(y) \;=\; \langle \mathcal{L}^A_{S_2y},\ \mathcal{R}^A_{S_2y} \rangle, \\
&where\ \mathcal{L}^A_{S_2y} \subseteq \mathcal{L}^A_\mathcal{T}\ and\ \mathcal{R}^A_{S_2y} \subseteq \mathcal{R}^A_\mathcal{T},\ is\ a\ subgraph \\
&at\ \rho_{k+1}(y) = rho_k(y).
\end{aligned}$$

Where we have Arg_{S2}(y) and Arg_{S2}(¬y), we have a *Single-point Debate about* y, $Arg_{S3}(y)$. The two graphs share only the literals $\{y, \neg y\}$, and no other rules or literals.

Definition 11. *Suppose two derived* AF*s, Arg*$_{S2}$*(y) =* $\langle \mathcal{L}^A_{S_2\neg y}, \mathcal{R}^A_{S_2\neg y}\rangle$ *and Arg*$_{S2}$*(y) =* $\langle \mathcal{L}^A_{S_2 y},\ \mathcal{R}^A_{S_2 y}\rangle$*:*

$$\begin{aligned}&A\ Single-point\ Debate\ about\ y\ is\\ &Arg_{S3}(y)\ =\ \langle \mathcal{L}^A_{S_2 y} \cup \mathcal{L}^A_{S_2\neg y}, \mathcal{R}^A_{S_2 y} \cup \mathcal{R}^A_{S_2\neg y}\rangle,\\ &where\ \mathcal{L}^A_{S_2\neg y} \cap \mathcal{L}^A_{S_2 y}\ =\ \{y, \neg y\}\\ &and\ \mathcal{R}^A_{S_2\neg y} \cap \mathcal{R}^A_{S_2 y}\ =\ \emptyset.\end{aligned}$$

Clearly a debate with subsidiary debates can be constructed to argue pro and con about other literals in the base debate; we start with a Arg$_{S2}$(y), then add further *Single-point Debates* about some literal in the graph other than y.

Example 5 shows the senses in a derived AF only with SI rules since they restrict the available preferred extensions.

Example 5. Suppose a Theory Base comprised of the rules (and related literals): r_7 : $x_6 \rightarrow \neg x_8, r_{10} : x_5, x_7 \rightarrow x_8, r_{11} : \neg x_3, x_4 \rightarrow x_7$. Figure 7 graphically represents the various senses of *argument* in an AF derived from this Theory Base.

In Figure 7, we have three subgraphs which represent an Argument; each Argument is derived from the corresponding rule of the Theory Base. For example $Arg_{S1}(\neg x_8)$, the argument for $\neg x_8$, is the graph comprised of nodes $\{\neg x_8, x_8, r_7, \neg x_6, x_6\}$ with the relations among them as given; the graph is derived from the rule of the Theory Base which corresponds to $r_7 : x_6 \rightarrow \neg x_8$. The other two rules of the Theory Base are also represented in the graph as subgraphs that represent an Argument. Figure 7 presents two Cases. The Case $Arg_{S2}(x_8)$ is derived from the following rules: r_{10} : $x_5, x_7 \rightarrow x_8, r_{11} : \neg x_3, x_4 \rightarrow x_7$. We see how the Arguments in the Case are linked; for instance, the graph of $r_{11} : \neg x_3, x_4 \rightarrow x_7$ has as claim x_7, which is the premise of $r_{10} : x_5, x_7 \rightarrow x_8$. The Case $Arg_{S2}(\neg x_8)$ is derived from the following rule (recall that an Argument can also be a Case): $r_7 : x_6 \rightarrow \neg x_8$. The Single-point Debate for x_8, $Arg_{S3}(x_8)$, is comprised of the Cases $Arg_{S2}(x_8)$ and $Arg_{S2}(\neg x_8)$.

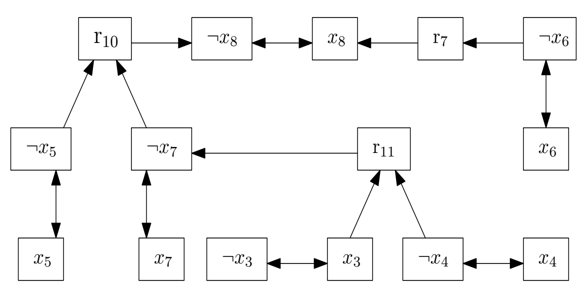

Fig. 7. Arguments, Cases, and Single-point Debates

In [4], there are some auxiliary definitions for *rebuttal*, *premise defeat*, and *undercutting* in this framework. However, space precludes presenting them here.

6 Concluding Remarks and Future Work

We have discussed in some detail comparison on one developed approach to argument instantiation [8] and noted other that remain to be compared in depth [7,2] though they share substantive similarities in terms of the Three Steps of Figure 1. Here we comment briefly on the somewhat different approach of *Abstract Dialectical Frameworks* (ADF) [9], which is presented as a generalisation of Dungian AFs but also as a means to represent instantiated arguments, e.g. logic programs [9]. Broadly, we may distinguish between approaches based on [1] that make use of nodes (arguments) and arcs (attacks) alone to determine extensions and those which use *auxiliary conditions to specify extensions with respect to successful attacks* such as preferences [8] or values [15]. The approach of [9] is a generalisation of the latter approach: in addition to nodes (which can be statements or literals) and links (generalised from arcs as attacks), there are acceptance conditions, which are functions for each statement from its parents (those nodes in a single link) to $\{in,out\}$. Given this generic approach to acceptance conditions, many complex aspects of argumentation can be accommodated. On the other hand, this emphasises reasoning with the (presumably correct) acceptance conditions rather than on the graph per se, which was one of the main advantages of the Dungian abstraction. For example, ADF remains to demonstrate that it abides by the Rationality Postulates or can generically reconstruct KBs. It adds the complexity of the acceptance conditions to the existing complexity of the graph [9]. On the other hand, our approach is compatible with ADF in the sense that given an AF derived from a KB, we can add auxiliary ADF acceptance conditions for other aspects of reasoning. In future, we plan to examine the advantages and disadvantages of the more specific approach to KB representation proposed here in comparsion with the more generic appoach of [9].

We have presented a method of instantiating a Theory Base which contains strict and defeasible rules in a Dung-style abstract argumentation framework, building on and refining [4]. The Theory Base is directly represented in the framework, and the conclusions of the Theory Base can be computed as extensions of that framework. Our method avoids the logic dependent step of generating arguments from the Theory Base and then organising them in a framework for evaluation. It does not introduce preferences or auxiliary means to determine successful arguments. The sceptically acceptable arguments of the framework are the consequences of the Theory Base under classical logic, assuming that the Theory Base is consistent: the consequences under a variety of non-monotonic logics can be identified as credulously acceptable arguments, with different non-monotonic logics corresponding to different ways of choosing between preferred extensions. We believe that this method provides a very clear way of instantiating Theory Bases as abstract argumentation frameworks. By separating the notion of a node from the ambiguous notion of argument, we have clear criteria for what constitutes a node in the framework. We can explain our reasoning in terms of arguments of the appropriate granularity. In addition, the variety of senses of "argument" emerge as structures within the framework, and can be used to explain the consequences.

In future work we will demonstrate the formal properties of our approach. In addition, we will further compare and contrast approaches to Theory Base instantiation in AFs. An important avenue of exploration and development is to add values, preferences, and weights to the KB, which then appear in the graph. In a different vein, we will explore the potential for improved explanation offered by our distinction between various senses of the term “argument”.

References

1. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2) (1995) 321–358
2. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artificial Intelligence **93** (1997) 63–101
3. Caminada, M., Amgoud, L.: On the evaluation of argumentation formalisms. Artificial Intelligence **171**(5-6) (2007) 286–310
4. Wyner, A., Bench-Capon, T., Dunne, P.: Instantiating knowledge bases in abstract argumentation frameworks. In: Proceedings of The Uses of Computational Argumentation. AAAI Fall Symposium (2009)
5. Dunne, P.E., Bench-Capon, T.J.M.: Coherence in finite argument systems. Artificial Intelligence **141**(1) (2002) 187–203
6. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming **4**(1) (2004) 95–137
7. Besnard, P., Hunter, A.: Elements of Argumentation. MIT Press (2008)
8. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation **1**(2) (2010) 93–124
9. Brewka, G., Woltran, S.: Abstract dialectical frameworks. In: Proceedings of the Twelfth International Conference on the Principles of Knowledge Represetnation and Reasoning (KR 2010). (2010) 102–211
10. Caminada, M., Wu, Y.: On the limitations of abstract argumentation. In Causmaecker, P.D., Maervoet, J., Messelis, T., Verbeeck, K., Vermeulen, T., eds.: Proceedings of the 23rd Benelux Conference on Artificial Intelligence, Ghent, Belgium (November 2011) 59–66
11. Wyner, A., Bench-Capon, T., Atkinson, K.: Three senses of “argument”. In Sartor, G., Casanovas, P., Rubino, R., Casellas, N., eds.: Computable Models of the Law: Languages, Dialogues, Games, Ontologies. LNAI 4884. Springer (May 2008) 146–161
12. Reiter, R.: A logic for default reasoning. Artificial Intelligence **13**(1-2) (1980) 81–132
13. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-Classical Logics **7**(1) (1997)
14. McCarthy, J.: Circumscription - a form of non-monotonic reasoning. Artificial Intelligence **13** (1980) 27–39
15. Bench-Capon, T.J.M.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation **13**(3) (2003) 429–448