Formalizing Balancing Arguments

Thomas F. GORDON [a,1], and Douglas WALTON [b]
[a] *Fraunhofer FOKUS, Berlin, Germany*
[b] *University of Windsor, Windsor, Canada*

Abstract. Dung intended his abstract argument frameworks to be used for modeling a particular form of human argumentation, where arguments attack each other and are evaluated following the principle summarized by "The one who has the last word laughs best." However this form does not fit a wide class of arguments, which is arguably more prototypical and common in human argumentation, namely arguments where pros and cons are balanced to choose among alternative options. Here we present a formal model of structured argument which generalizes Dung abstract argumentation frameworks to also handle balancing. Unlike most other models of structured argument, this model does not map structured arguments to abstract arguments. Rather it generalizes abstract argumentation frameworks, allowing them to be simulated using structured arguments. The model can handle cumulative arguments ("accrual") without causing an exponential blowup in the number of arguments and has been fully implemented in Version 4 of the Carneades Argumentation System.

Keywords. structured argumentation, argument evaluation, argument accrual, cumulative arguments, balancing arguments

1. Introduction

A wide class of human argumentation involves the balancing of pros and cons to choose among a set of options:

- Practical argumentation to choose a course of action involves balancing the costs and benefits of alternative actions, taking into consideration multiple-criteria, in addition to arguing about the preconditions and effects of the actions. Value-based practical reasoning [2] evaluates costs and benefits relative to particular audiences, in terms of the degree to which each action promotes or demotes some value. Arguing about governmental policies is of this type [9].
- Theoretical argumentation, both in natural science and in the humanities, including law, involves constructing, comparing and choosing among alternative theories, taking into consideration multiple evaluation criteria, such as the extent to which the theories explain the evidence or, in the law, precedent cases, and their simplicity, in line with Occam's razor, among other factors, to choose the most coherent theory.

[1] Corresponding Author: Prof. Dr. Thomas F. Gordon, Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, Berlin, Germany; E-mail: thomas.gordon@fokus.fraunhofer.de.

- Factual argumentation about whether or not some event occurred, involves the acquisition and weighing of evidential arguments, such as witness testimony, as well as the construction, comparison and balancing of the properties of alternative narratives ("stories"), to choose the most coherent narrative [4].
- Arguing about whether some concrete fact situation can or should be subsumed under some abstract concept, for example whether or not users have a "reasonable expectation of privacy" regarding the personal data on their smart phones, which would then be protected from "unreasonable searches and seizures" by the Fourth Amendment of the US Constitution. This can involve balancing different methods of constitutional interpretation (e.g. literal, historical and teleological). Moreover teleological interpretation can involve the balancing of interests in a way which respects the balance ("proportionality") set by the founders in the constitution.

The leading computational model of argument, abstract argumentation frameworks [8], was not designed to handle balancing arguments, but rather another form of argumentation, where arguments are viewed as attacking arguments and evaluated according to the following principle:

> The goal of this paper is to give a scientific account of the basic principle "The one who has the last word laughs best" of argumentation, and to explore possible ways for implementing this principle on computers. [8, pg 322]

Most of the leading models of structured argumentation [3,12,17] are defined as preprocessors for an argument evaluator for abstract argumentation frameworks, following the methodology proposed by Dung in [8, pg 348]:

> Any argumentation system is composed of two essential components: One for generating the arguments together with the attack-relationship between them. The other is for determining the acceptability of arguments. So we can think of an argumentation system as consisting of two units, an argument generation unit, AGU, and an argument processing unit, APU.

Dung illustrated this process with a pipeline, where the AGU generates an argumentation framework (arguments and attacks) and the APU then evaluates this framework to determine which of these arguments are acceptable. In practice, structured models of argument have extended this pipeline with an additional process at the end, for labeling the *statements* (propositions) in the structured model of argument. Typically, a statement is acceptable (in) if and only if it is supported by an acceptable argument.

The linearity of this pipeline presents a problem when one wants to model balancing arguments, since the value (weight) of a balancing argument can depend on the acceptability of statements and, recursively, the acceptability of statements depends, as in the extended Dung methodology, on the value the arguments supporting them. When balancing, pro and con arguments are weighed against each other. An out premise can reduce (or increase!) the weight of an argument, without defeating it completely.

It is unclear whether it would be possible in principle to define a model of structured argumentation for balancing arguments using Dung's pipeline methodology. Since there are no limits on the structure of arguments or the attack relation used to instantiate the abstract framework, some clever encoding of balancing arguments may be possible, but one has to wonder how straightforward or intuitive such a model of balancing arguments could be. In this paper we take a more direct, requirements-driven approach to modeling structured argumentation with support for both attacks and balancing, which preserves the recursivity of the balancing process, without worrying about trying to find some way to model this process as a linear pipeline to comply with Dung's methodology.

2. The Formal Model

2.1. Structure

Let $\mathscr{L}$ be a logical language for expressing statements (propositions). As in ASPIC+ [14], this model is a "framework". It can be instantiated with any logical language.

An *argumentation scheme* is an abstract structure in this framework providing functions for generating, validating and weighing arguments. The framework can be instantiated with various models of argumentation schemes. For our purpose here of modeling balancing arguments, only the weighing functions of argumentation schemes are relevant. See Definition 5 for the signature and further details of weighing functions.

Definition 1 (Argument) *An* argument *is a tuple* (s,P,c,u)*, where:*

- *s is the scheme instantiated by the argument*
- *P, the* premises *of the argument, is a finite subset of* $\mathscr{L}$
- *c, a member of* $\mathscr{L}$*, is the* conclusion *of the argument, and*
- *u, a member of* $\mathscr{L}$*, is the* undercutter *of the argument.*

This model of argument closely fits the usual conception of an argument in informal logic and argumentation theory in philosophy [18]. Notice that an argument here, unlike in ASPIC+, is not a complete proof tree, but rather only a single inference step in such a proof tree. Undercutters here are modeled in the same way as in ASPIC+, with a proposition in $\mathscr{L}$ for each undercutter. In practice, these propositions will typically be constructed by applying some predicate to a term naming the argument, such as undercut(a_1). But this is a detail to be worked out when instantiating the framework. We also call arguments undercutters which have undercutter statements as their conclusion. Notice that the argument includes a reference to the scheme used to construct (or reconstruct) the argument. This will be used to weigh the argument.

Example 1 *Following the tradition of [5], let us use as our running example a practical reasoning task about choosing a car to buy. Let us assume that a domain-dependent argumentation scheme for car buying has been defined, where the premises express the claimed properties of a particular car, one for each of the criteria to be considered, and the weighing function of the scheme computes a weighed sum of the* proven *(not claimed) properties of the car, where the weight assigned to each property by the scheme is chosen to reflect the relative importance of the criterion, relative to the other criterion, in the manner of multi-criteria decision analysis. Here is an example of an argument for a particular auto, applying this scheme:*

Let $a_1 = (s,P,c,u)$ *be an argument for buying a Porsche, where:*

- *s is a car buying scheme, described in more detail in Example 4*
- *P, the premises, are:*
    1. *type(porsche,sports)*

2. *price(porsche,high)*
3. *safety(porsche,medium)*
4. *speed(porsche,fast)*

- *c, the conclusion, is buy(porsche), and*
- *u, the undercutter, is undercut*(a_5)

Definition 2 (Issue) *An* issue *is a tuple* (O, f)*, where:*

- *O, the* options *(also called* positions*) of the issue, is a finite subset of* $\mathscr{L}$.
- f*, the* proof standard *of the issue, is a function which tests whether an option satisfies the standard. See Definition 6.*

Issues are inspired by Issue-Based Information Systems (IBIS) [11]. They extend the concept of a "contrary" in the ASPIC+ model of structured argument, from a binary relation to an n-ary relation. Allowing more than two options is important for two reasons:

1. To allow more than two alternative options in deliberation dialogues and other decision-making contexts.
2. To avoid false dilemmas, by allowing alternatives other than true or false (or yes or no) for issues representing questions of the kind "Have you stopped beating your spouse?".

Proof standards of issues are borrowed from the 2007 version of Carneades [10]. Associating proof standards with issues is designed to assure that the same proof standard applies to every position of the issue.

Definition 3 (Argument Graph) *An* argument graph *is a tuple* (S, A, I, R)*, where:*

- *S, the* statements *of the argument graph, is a finite subset of* $\mathscr{L}$.
- *A, the* assumptions*, is a subset of S assumed to be provable.*
- *I, the* issues *of the argument graph, is a finite set of issues, where every position of every issue is a member of S and no* $s \in S$ *is a position of more than one* $i \in I$*, and*
- *R, the* arguments *of the argument graph, is a finite set of arguments, where all conclusions, premises and undercutters are members of S.*

These structures are called graphs for historical reasons. Admittedly this a bit of an abuse of terminology. But every argument graph (S, A, I, R) can be easily mapped to a directed graph (V, E) as follows:

- The vertices, V, of the graph consist of the statements (S), issues (I) and arguments (R) of the argument graph.
- The edges, E, of the graph are constructed by linking arguments in A to their premises, conclusions and undercutters in S, and issues in I to their options in S, in the obvious way.

In most other models of structured argument, argument graphs for structured arguments are not formally defined. In [3], Besnard and Hunter use the term "argument graph" as a synonym for abstract argumentation frameworks. In ASPIC+ arguments are proof trees. Sets of such arguments are often visualized in ASPIC+ presentations as an argument graph, where each argument is a subgraph of the argument graph, but the argument graph per se is not a part of the formal ASPIC+ model.

Example 2 *Figure 1 shows an argument graph for the car buying example, with an argument for buying a Porsche and another argument for buying a Volvo. The labels of the statement nodes, displayed with colors, and arguments, displayed as numbers (weights) on the edges from the arguments to their conclusions, are explained in Section 2.2. The proof standard "PE" used by both issues, means "preponderance of the evidence" and is also defined in Section 2.2. Undercutters are visualized with dashed edges.*

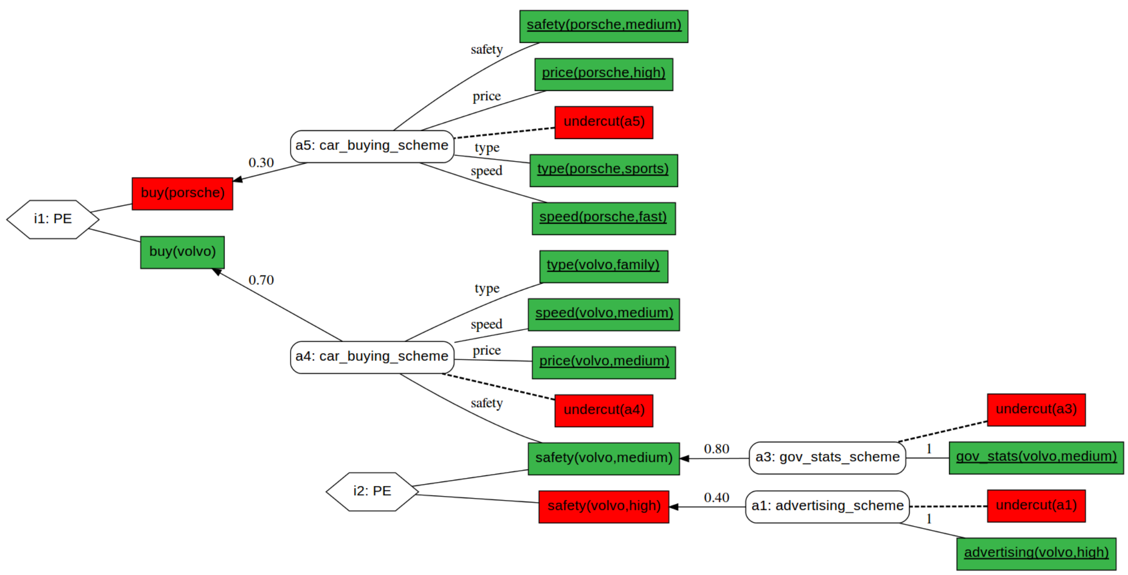

Figure 1. Example Argument Graph

2.2. Semantics

The semantics of argument graphs is defined here in a way inspired by and analogous to the labeling semantics of abstract argumentation frameworks [1], but without mapping argument graphs to abstract argumentation frameworks.

Definition 4 (Labeling) *A* labeling *is a total function from $\mathscr{L}$ to $\{$**in**$,$**out**$,$**undecided**$\}$.*

Notice that *statements*, not arguments, are labeled **in**, **out**, or **undecided** here, unlike the labeling semantics for abstract argumentation frameworks. Arguments here are labeled by their weights, as described below.

Example 3 *In the argument map shown in Figure 1, the statements shown in green and red are labeled* ***in*** *and* ***out****, respectively. All other statements in $\mathscr{L}$ are, by default, labeled* ***undecided****.*

Now we are in position to define weighing functions of argumentation schemes more precisely.

Definition 5 (Weighing Function) *A* weighing function *maps a* $(labeling \times argument\ graph \times argument)$ *tuple to normalized weights, real numbers in the range of 0.0 to 1.0. Every weighing function must assign the weight of 0.0 to an argument in a labeling if its undercutter is* ***in*** *in the labeling.*

Notice that the weight of an argument can depend on:

- the labeling
- properties of the argument graph, including but not limited to properties of other arguments about the same issue
- properties of the argument, such as the scheme applied

It is the potential dependence of the weight of an argument on the labeling of statements in the argument graph which makes it unclear how this model could be mapped to the pipelined evaluation model of abstract argumentation frameworks, where the labeling of all statements takes place at the end of the pipeline, after all the arguments have been labeled. Not all weighing functions may be sensible. An interesting project for future work might be to define further rationality constraints for weighing functions, in addition to assuring that undercut arguments weigh 0.0.

Example 4 *In the example shown in Figure 1, argument* a_4 *applies the domain-dependent car buying argumentation scheme. The weighing function of this scheme computes a weighted sum of the proven (not claimed) properties of the option supported by the argument, in the manner of multi-criteria decision analysis. In the example, an issue has been made out of the premise of* a_4 *stating that Volvos have medium safety, in issue* i_2*. Had this issue been resolved in favor of the other position of the issue, claiming that Volvos are not merely medium safe, but rather highly safe, then the argument for buying the Volvo,* a_4*, would have weighed more than it does,* 0.7*. This illustrates that the failure of a premise can not only weaken an argument, but also strengthen it,* a fortiori*. The example also illustrates how the weight of an argument,* a_4*, can depend on the label of a statement, safety(volvo,medium), which in turn (recursively) depends on weights of other arguments,* a_1 *and* a_3*.*

Definition 6 (Proof Standard) *A* proof standard *is a mapping from* $(labeling \times argument\ graph \times statement)$ *to* $\{\mathbf{true}, \mathbf{false}\}$*. A statement s* satisfies *a proof standard, f, given a labeling l and argument graph AG, iff* $f(l, AG, s) = \mathbf{true}$*. Since proof standards are used to justify decisions, a proof standard may allow at most one position of an issue to satisfy the standard.*

Example 5 *The* preponderance of evidence *proof standard can be defined as follows: a position of an issue satisfies the preponderance of evidence standard in an argument graph AG, if and only if there exists an argument in AG for this position (i.e. having this position as its conclusion) which weighs more than every argument in AG for every other position of the same issue, where the weight of an argument,* a_i*, is derived by applying the weighing function of the argumentation scheme of* a_i *to* (l, AG, a_i)*.*

Definition 7 (Applicable Argument) *An argument* $r \in R$ *is* applicable *in a labeling l if and only if:*

- *The undercutter of r is* ***in*** *or*

- *The undercutter of r is* ***out*** *in l and every premise of r is not* ***undecided*** *in l.*

Notice that premises of an argument need not be **in** for the argument to be applicable. Premises that are **out** can weaken or strengthen the argument, without causing it to become inapplicable. Also, somewhat unintuitively, an argument can be applicable even if its undercutter is **in**. Undercut arguments have zero weight. (See Definition 5.)

Example 6 *In the argument graph shown in Figure 1, all of the arguments are applicable, since all of their undercutters are* ***out*** *and none of their premises are* ***undecided****.*

Definition 8 (Supported Statement) *In a labeling l, a statement s is* supported *by an argument graph* (S,A,I,R) *iff there exists an argument* $r \in R$ *such that*

- *s is the conclusion of r,*
- *r is* applicable *in l, and*
- $w(l,AG,r) > 0.0$*, where w is the weighing function of the scheme of r.*

In other words, a statement is *supported* if it is the conclusion of an applicable argument weighing greater than 0.0. Note that a supported statement is not necessarily labeled **in** in *l*.

Example 7 *In the argument graph shown in Figure 1 several statements are supported, including safety(volvo,medium), safety(volvo,high), buy(volvo) and buy(porsche).*

Definition 9 (Unsupported Statement) *Let l be a labeling,* (S,A,I,R) *be an argument graph, and P be the subset of the arguments R having a statement s as their conclusion. s is* unsupported *by the argument graph iff*

- *P is empty or*
- *for every argument* $r \in P$*: r is applicable in l but the weight of r in l is 0.0, i.e.* $w(l,AG,r) = 0.0$*, where w is the weighing function of the scheme of r.*

That is, a statement is *unsupported* if every argument for this statement (i.e. having this statement as its conclusion) is applicable but with a weight of 0.0. Note that supported and unsupported are not duals: A statement can be neither supported nor unsupported.

Definition 10 (Resolvable Issue) *An issue i is* resolvable *in a labeling l, if for every position p of i: every argument* $r \in R$ *with the conclusion p is applicable in l in l.*

The basic intuition here is that an issue in an argument graph is ready to be resolved in a labeling, if the labeling provides enough information to evaluate every argument for every position of the issue. It may be that no position of a resolvable issue satisfies its proof standard. Thus being resolvable does not imply that some position of the issue is **in**.

Example 8 *Both issues of the argument graph shown in Figure 1 are resolvable.*

Definition 11 (Conflict Free Labeling) *Let AG be an argument graph* (S,A,I,R)*. A labeling l is* conflict free *with respect to AG iff, for every statement* $s \in S$*:*

- *if $s \in A$ then $l(s) \neq$ **out***
- *if $s \notin A$ and s is* unsupported *in l then $l(s) \neq$ **in***
- *if s is not a position of some issue $i \in I$ and s is* supported *in l then $l(s) \neq$ **out***
- *if s is a position of some issue $i \in I$ such that i is resolvable in l and s does not satisfy the proof standard of i then $l(s) \neq$ **in***
- *if s is a position of some issue $i \in I$ such that i is resolvable in l and s satisfies the proof standard of i then $l(s) \neq$ **out***

The concept of conflict-freeness here is analogous to conflict-freeness in abstract argumentation frameworks. The purpose is to define constraints which must be satisfied by every labeling of an argument graph. The constraints tell us what the labels may not be, but do not tell us what they must be. Labeling a statement **undecided** is always permitted. So, more precisely, the constraints tells us when a statement may not be **in** or **out**:

- Assumptions may not be **out**.
- An unsupported statement which is not an assumption may not be **in**.
- If a supported statement is not at issue, it may not be **out**.
- If an issue is resolvable and some position of the issue does not satisfy the proof standard of the issue, then the position may not be **in**.
- If an issue is resolvable and some position of the issue satisfies the proof standard of the issue, then the position may not be **out**.

Inspired also by abstract argumentation frameworks, we define the semantics of argument graphs using fix-points of a characteristic function:

Definition 12 (Characteristic Function) *Let AG be an argument graph (S,A,I,R). The* characteristic function *of argument graphs, f : labeling $\rightarrow$ labeling, is defined as follows:*

$f(l)$ =
  let m be the resulting labeling
  for each $s \in S$:
    if $l(s) \neq$ **undecided** then $m(s) = l(s)$
    else if $s \in A$ then $m(s) =$ **in**
    else if s is unsupported in l
      then $m(s) =$ **out**
    else if s is not a position of some issue and s is supported in l
      then $m(s) =$ **in**
    else if s is a position of some issue $i \in I$ such that
      i is resolvable in l and s does not satisfy the proof standard of i
      then $m(s) =$ **out**
    else if s is a position of some issue $i \in I$ such that
      i is resolvable in l and s satisfies the proof standard of i
      then $m(s) =$ **in**
    else $m(s) = l(s)$

The basic intuition behind this characteristic function is that it is intended to complete a labeling of an argument graph, relabeling some or all **undecided** statements to **in** or **out**, as much as possible in a "single step". The characteristic function can be applied repeatedly (iteratively) until a fix-point is found, i.e. where $f(l) = l$.

Fix-point semantics requires the characteristic function to be monotonic:

Definition 13 (In and Out Statements of a Labelling; Extensions) *Given an argument graph* (S, A, I, R) *and a labeling* l*, let* $i(l)$*, called the* extension *of the argument graph in* l*, denote the subset of* S *labeled* ***in*** *in* l *and* $o(l)$ *denote the subset of* S *labeled* ***out*** *in* l*.*

Conjecture 1 (Monotonicity of the Characteristic Function) *Let us overload* $\subseteq$ *to also denote a preorder on labelings, where* $l_1 \subseteq l_2$ *iff* $i(l_1) \subseteq i(l_2)$ *and* $o(l_1) \subseteq o(l_2)$*. The characteristic function* f *is monotonic, preserving this order: for every labeling* l_1 *and* l_2*, if* $l_1 \subseteq l_2$ *then* $f(l_1) \subseteq f(l_2)$*.*

Finally, assuming the monotonicity conjecture is true, we can define various fix-point semantics of argument graphs, in a way analogous to the semantics of abstract argumentation frameworks:

Definition 14 (Fix-Point Semantics) *Given an argument graph* (S, A, I, R)*, a labeling* l *is:*

- admissible *if and only if* l *is conflict-free.*
- complete *if and only if* l *is admissible and* $f(l) = l$*, i.e.* l *is a fix-point of* f*.*
- grounded *if and only if* l *is complete and minimal, i.e. there does not exist a labeling* l' *such that* $l' \subset l$*.*
- preferred *if an only if* l *is complete and maximal, i.e. there does not exist a complete labeling* l' *such that* $l' \supset l$*.*

Example 9 *The grounded labeling of the argument graph of the running example is shown in Figure 1. The* ***in*** *and* ***out*** *labels of statements are shown by filling the boxes of the statements with green and red color, respectively. (No statements are* ***undecided*** *in the grounded extension of this argument graph.)*

We are developing a version of this formal model in Higher-Ordered Logic (HOL) for the Isabelle proof assistant[2]. The Isabelle version of the formalization is available online [3]. We plan to use Isabelle to help us prove properties of the model, in future work, including Conjecture 1, about the monotonicity of the characteristic function.

The formal model has been fully implemented in Version 4 of the Carneades Argumentation System. Carneades is open source software, published using the MPL 2.0 license.[4] Carneades can be used as a command line program or as a web application. You can try out the web version online using the Carneades server.[5]

3. Related Work

This formal model of structured argument has been clearly inspired by Dung's work [8], even if we have chosen to not follow his recommended pipeline methodology by trying

[2]https://isabelle.in.tum.de
[3]https://github.com/carneades/caes2-formalization
[4]https://github.com/carneades/carneades-4
[5]http://carneades.fokus.fraunhofer.de/carneades

to map argument graphs to abstract argumentation frameworks. Rather, we have used Dung's approach, in particular its use of fix-point semantics, as a model and adapted it to the purpose of handling balancing arguments, in addition to attack relations among arguments. Some parts of [8] suggest that Dung intended abstract argumentation frameworks to be expressive enough for evaluating all kinds of human argumentation, but all of his examples were from computer science, nonmonotonic (defeasible) logic and logic programming. He did not consider how to model balancing arguments, which are widespread in human argumentation.

We conjecture that it is possible, indeed straightforward, to simulate abstract argumentation frameworks with the model of argument graphs presented here. An example explaining how this can be done can be found online.[6] Both arguments and attacks of the abstract framework are mapped to structured arguments. Attacks of the abstract framework are modeled as undercutters. If the abstract framework has m arguments and n attacks, the resulting argument graph has at most $2*m$ statements and $m+n$ arguments. Thus the simulation has polynomial complexity.

Another source of inspiration for this work was ASPIC+ [14]. All three kinds of attack relations supported by ASPIC+ (premise attacks, rebuttals and undercutters) are also supported in our model. We have successfully reconstructed many of the examples used to illustrate ASPIC+. These examples are available online.[7]

In future work we would like to show formally how to simulate ASPIC+ using our model. We conjecture our model is both simpler and more expressive than ASPIC+, considered as a whole, despite some elements of our model being more complex. Both models are frameworks which can be instantiated in various ways (e.g. logical language, priority relation over arguments), but only our system can handle balancing arguments. ASPIC+ can handle a special case, argument accrual, but only at the cost of replacing each accrued argument with multiple arguments, one for each subset of its premises, causing an exponential blow-up in the number of arguments, which negatively impacts both on the efficiency of argument evaluation and the comprehensibility of argument maps used to visualize and explain the evaluation. In [13], Prakken defined three principles of argument accrual, including the principle that accrued arguments can be weaker than arguments with subsets of their premises. We conjecture that these principles are satisifed by the model presented here, but this remains to be formally proved.

We considered basing this model on Abstract Dialectical Frameworks (ADFs) [6], since they provide a convenient platform for defining a wide variety of graph-based formalisms. The nodes of ADFs can in principle model anything, not just arguments, including presumably also statements and issues, as we need. However, ADFs evaluate and label nodes using functions attached to nodes which depend only upon the parents of the nodes, i.e. the immediate predecessors of the node in the directed graph. This does not appear to be general enough for our purposes, as can be seen in the running example used here, where the weighing function of the car buying scheme needs to consider not only the premises of the argument, but also alternative positions of each premise at issue, since the weight of the argument depends on the proven properties of the car being considered, not only its claimed properties. These other positions are three links away in the argument graph from the argument being weighed.

[6] https://github.com/carneades/carneades-4/blob/master/examples/AGs/YAML/dung-attack-cycle.yml
[7] https://github.com/carneades/carneades-4/tree/master/examples/AGs/YAML

Finally, of course the model presented here is derived from our own prior work on structured argumentation [10] and preserves all of its features, including its support for variable proof standards and its support for modeling the two kinds of critical questions of argumentation schemes, using assumptions and exceptions. However the new model is simpler and more general in several ways:

1. Con arguments and rebuttals are now modeled as arguments pro other positions (options) of issues.
2. There is now only one kind of premise, instead of three (ordinary, exception, assumption). Assumptions are now a subset of the statements of the argument graph. Exceptions are now modeled using undercutters, which are more general, since an undercutter can have more than one premise.
3. All premises are positive. (Previously, premises could be positive or negative.)
4. The new model lifts the restriction to cycle-free argument graphs, thanks to its Dung-inspired fix-point semantics.
5. Argument weights are now derived, by applying weighing functions attached to argumentation schemes, rather than asserted.

The main additional complexity in the new model is its introduction of a third node type for issues, in addition to statements and arguments.

One important advantage of the new formal model is that argument graphs are now much closer to the conceptual model underlying the argument diagrams typically used in informal logic textbooks, such as [18]. This conceptual model underlies several argument mapping tools, including Araucaria [16], and is also the basis for the Argument Interchange Format (AIF) [15]. Version 4 of Carneades, based on the new model presented here, can import and evaluate AIF files.

4. Conclusion

This paper has presented an original formal model of structured argument with support for both attack relations among arguments (premise defeat, rebuttals and undercutters) as well as balancing arguments, using argument weighing functions. The model has been illustrated using a practical reasoning example about which car to buy, where the weighing function computes a weighted sum of the proven properties of the proposed options, in the style of multi-criteria decision analysis. This model can handle cumulative arguments [19] and argument accrual [13] without causing an exponential blow-up in the number of arguments. While the model does not map structured arguments to abstract arguments, it is inspired by the fix-point semantics of abstract argumentation frameworks and uses comparable methods to handle and resolve cycles in argument graphs. The formal model has been fully implemented in Version 4 of the Carneades argumentation system, for grounded semantics. Many examples from the literature on structured argumentation have been successfully reconstructed, and several new examples have been developed to illustrate the model's features for balancing arguments.

In future work we would like to formally prove Conjecture 1 about the monotonicity of the characteristic function, as well proving the conjecture that the model can simulate abstract argumentation frameworks, for common semantics (e.g. complete, grounded, preferred) and formally investigating relationships between this model and other models of structured argument, in particular ASPIC+. We also plan to investigate whether or not Caminada's rationality postulates for structured argumentation [7], e.g. closure, direct consistency and indirect consistency, are meaningful in the context of this model and, if so, whether they are satisfied. We plan to use the version of the formal model in Higher-Order Logic for the Isabelle proof assistant to facilitate this future work.

References

[1] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[2] Trevor Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[3] Philppe Besnard and Anthony Hunter. Constructing argument graphs with deductive arguments: A tutorial. *Argument and Computation*, 5(1):5–30, 2014.

[4] Floris J. Bex. *Arguments, Stories and Criminal Evidence: A Formal Hybrid Theory*. Springer, Dordrecht, 2011.

[5] Gerhard Brewka and Thomas F Gordon. How to buy a porsche: An approach to defeasible decision making. In *Working Notes of the AAAI-94 Workshop on Computational Dialectics*, pages 28–38, Seattle, Washington, 1994.

[6] Gerhard Brewka and Stefan Woltran. Abstract Dialectical Frameworks. In *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*, pages 102–111. AAAI Press, 2010.

[7] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, April 2007.

[8] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[9] Isabela Fairclough and Norman Fairclough. *Political discourse analysis: A method for advanced students*. Routledge, 2013.

[10] Thomas F. Gordon, Henry Prakken, and Douglas Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10-11):875–896, 2007.

[11] Werner Kunz and Horst W.J. Rittel. Issues as elements of information systems. Technical report, Institut für Grundlagen der Planung, Universität Stuttgart, 1970.

[12] Sanjay Modgil and Henry Prakken. The ASPIC+ framework for structured argumentation: A tutorial. *Argument and Computation*, 5(1):31–62, 2014.

[13] Henry Prakken. A Study of Accrual of Arguments, with Applications to Evidential Reasoning. In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law*, pages 85–94, New York, 2005. ACM Press.

[14] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1:93–124, 2010.

[15] I. Rahwan and Chris Reed. The Argument Interchange Format. In I. Rahwan and Chris Reed, editors, *Argumentation in Artificial Intelligence*. Springer, 2009.

[16] Chris A Reed and Glenn W A Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal of AI Tools*, 13(4):961–980, 2004.

[17] Francesca Toni. A tutorial on assumption-based argumentation. *Argument and Computation*, 5(1):89–117, 2014.

[18] Douglas Walton. *Fundamentals of Critical Argumentation*. Cambridge University Press, Cambridge, UK, 2006.

[19] Douglas N. Walton, Christopher W. Tindale, and Thomas F. Gordon. Applying recent argumentation methods to some ancient examples of plausible reasoning. *Argumentation*, 28(1):85–119, 2014.