# Explaining the Outcome of Knowledge-Based Systems; a discussion-based approach

**Martin Caminada**[1] and **Mikołaj Podlaszewski**[2] and **Matt Green**[3]

**Abstract.** Many inferences made in everyday life are only valid in the absence of explicit counter information. This has led to the development of nonmonotonic logics. The kind of reasoning performed by these logics can be difficult to explain to the average end-user of a knowledge based system that implements them. Although the system can still give advice, it is hard for the user to assess the rationale behind this advice. In this paper we propose an argumentation approach that enables the advice to be assessed through an interactive dialogue with the system much like the discussion one might have with a colleague. The aim of thie dialogue is for the system to convince the user that the advice is well-founded.

## 1 NONMONOTONIC REASONING

Human-style common sense reasoning is inherently nonmonotonic. When new information becomes available, some of our previous beliefs and inferences might no longer be warranted. An often cited example of philosopher and AI researcher John Pollock is that from the fact that an object looks red one might reasonably infer that the object really is red. However, if one later obtains the additional information that the object was in fact illuminated by a red light, one should block the conclusion that the object really is red (unless one also has other reasons to believe so). This kind of reasoning contrasts strongly with the approach taken by formalisms like classical logic. Here, the notion of entailment is essentially monotonic, meaning that whenever one adds new facts, one can only obtain more (possibly the same) and never fewer conclusions.[4]

The need for nonmonotonic reasoning (NMR) comes from the fact that many inferences made in everyday life are defeasible. That is, they are only valid in the absence of explicit counter information. The need to accommodate this type of reasoning in formal logic has led to the field of nonmonotonic logics, of which Default Logic and Circumscription are some well-known examples.

One of the purposes of nonmonotonic logics was to be implemented in knowledge-based systems, which would then be able to assist its users in things like diagnosis and decision making. One of the difficulties, however, was that the kind of reasoning performed by nonmonotonic logics can be notoriously difficult to explain to the average end-user, who has no explicit background in how these formalisms function. Although the resulting system could still give advice, it would be hard for the user to assess how this advice came about, why it is indeed the right advice and whether any objections that the user might have are indeed taken into account. That is, the challenge would be for the system to *justify* its advice in a way that can actually be understood by the user. The failure to address this issue could be seen as one of the reasons why the field of nonmonotonic logics did not manage to come up with any widely used commercial applications.[5]

## 2 ARGUMENT-BASED REASONING

A new impulse was given in the 1990s, with the development of formalisms for argument-based reasoning (see for instance [17, 13, 14, 19]) which culminated in the landmark paper of Dung [6]. In this paradigm, an argument is essentially an aggregation of reasons that, when taken together, supports a particular conclusion. An argument can be attacked by other arguments (like the argument "the object is red because it looks red" is attacked by the argument "the object is illuminated by a red light, so the fact that it looks red is not a reason for it actually being red"). The idea is that, given the information that is available, one can construct the relevant arguments and examine which arguments attack which other arguments. The result can be visualised in a graph, in which the nodes represent arguments and the arrows represent the attack relation. Given such an *argumentation framework* (as this graph is called in [6]) one should then determine (using a formal criterion) which of the arguments should be accepted, rejected or abstained from having an explicit opinion about. As an example, consider the following hypothetical situation involving three arguments:
$A$: "We should give the patient aspirin, because he's in pain."
$B$: "We should not give him aspirin, because he's diabetic and existing research indicates that providing aspirin leads to complications for patients who are diabetic."
$C$: "The research on which this claim is based has been proven to be flawed and has been refuted by clinical evidence."

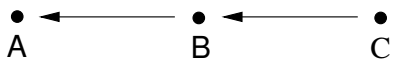


**Figure 1.** Simple argumentation framework

This situation is graphically depicted in the graph in Figure 1,

[1] University of Aberdeen, Scotland
[2] University of Luxembourg, Luxembourg
[3] University of Aberdeen, Scotland, email: mjgreen@abdn.ac.uk
[4] Formally, when $\Phi_1$ and $\Phi_2$ are sets of formulas in a particular logic of which $Cn$ stands for the consequence relation, then monotonicity means that if $\Phi_1 \subseteq \Phi_2$ then $Cn(\Phi_1) \subseteq Cn(\Phi_2)$.
[5] One notable exception is Answer Set Programming (ASP). ASP, however, is mainly aimed at providing efficient computation for problems involving constraint satisfaction, instead of tackling the original NMR challenge of how to reason with rules of thumb that are subject to exceptions.

where the nodes represent arguments and the arrows represent attacks between the arguments. Here, the idea is that at least argument $C$ should be accepted, because it is not attacked by any other argument. Argument $B$, however, should be rejected because it is attacked by an argument ($C$) that is accepted. Argument $A$ is the most interesting case. Its only attacker ($B$) is rejected and therefore cannot be a valid ground anymore against the acceptance of $A$. Since there is no other argument attacking $A$, $A$ should therefore be accepted. This is in line with the general approach of formal argumentation: an argument is accepted unless there are valid grounds against doing so. In this simple example it can be seen that the status of an argument depends on the status of its attackers, which in its turn depends on the status of their respective attackers, etc. However, in more complex graphs (especially those containing cycles) things are not that obvious. In that case, a formal criterion for acceptance and rejection (called an "argumentation semantics") is needed. An example of such a criterion is that of a *complete labelling* [1, 3]. Here, the idea is to assign each argument exactly one label, which can be `in` (indicating acceptance), `out` (indicating rejection) or `undec` (indicating that there are insufficient grounds for either acceptance or rejection).

Existing results in formal argumentation theory state that each graph ("argumentation framework") has at least one complete labelling (that is, an assignment of `in`, `out` and `undec` that satisfies the above three conditions). If the graph contains cycles, more than one complete labelling can exist. Informally, the concept of a complete labelling can be seen as a reasonable position one can take based on the conflicting information encoded in the argumentation framework.

The popularity of argument-based inference formalisms[6] can partly be explained by the facts that:

1. these have been shown to be powerful enough to model a wide range of existing formalisms for nonmonotonic reasoning (like Default Logic [16] and logic programming under various semantics [7, 8, 18]),
2. efficient proof procedures and algorithms are available, and
3. formal argumentation can be seen as a step forward to making formal nonmonotonic inference understandable to end-users

The traditional approach to formal argument-based inference consists of a three-step process. The first step is, given a particular knowledge base, to construct the relevant arguments and examine how they attack each other (that is, to construct the argumentation framework). The second step is to evaluate the resulting argumentation framework (for instance, to determine the complete labellings). The third step is then to examine what this means at the level of conclusions (recall that each argument has at least one conclusion). That is, for each complete labelling of arguments, one determines the associated complete labelling of conclusions (for instance by applying the procedure described in [20]).

One of the things that is missing in the above process is the dialectical aspect. The traditional argumentation process (as for instance formalised in ASPIC [2, 15, 11]) aims at putting all arguments on the table and then simply computing which of them should be accepted. However, in natural argumentation one also encounters the concepts of dialogue and discussion. Where do these concepts fit in when it comes to formal argumentation theory?

[6] Another application of argumentation theory can be found in the field of game theory (see [6] for details). However, in the current paper we focus on argumentation for (nonmonotonic) inference.

## 3 ARGUMENTATION AS DIALOGUE

A complete labelling can be achieved by assigning a single label to each argument. The following rules show the possible labels:

1. if the argument is labelled `in` (accepted) then all its attackers have to be labelled `out` (rejected)
2. if the argument is labelled `out` (rejected) then it has at least one attacker that is labelled `in` (accepted)
3. if the argument is labelled `undec` (abstained) then not all its attackers are labelled `out` (so there are insufficient grounds to accept it) and it doesn't have an attacker that is labelled `in` (so there are insufficient grounds to reject it)

A dialogue game consists of the following moves:

**claim** This is the first move in the dialogue. The proponent claims that a particular argument has to be labelled `in`. This creates a commitment that the proponent enters into.

**why** The opponent asks why a particular claim holds – why a particular argument has to be labelled a particular way

**because** A party explains why the label of a particular argument has to be the way it was earlier claimed to be.

**concede** With this move, a party concedes part of the statements uttered earlier by the other party

A dialogue takes place under the following rules:

- The proponent (P) and the opponent (O) take turns. Each turn of P consists of a single move: `claim` or `because`. O plays one or more moves in a turn. O's turn starts with an optional sequence of `concede` moves and finishes (when possible) with a single `why` move.
- P gets committed to arguments used in `claim` and `because` moves; O gets committed to arguments used in `concede` moves.
- P starts with `claim in` $(A)$ where A is the main argument of the discussion: `claim` cannot be repeated later in the game.
- In consecutive turns P provides reasons for the directly preceding `why` ($\mathcal{L}$) move of O by moving `because` ($\mathcal{L}'$) where $\mathcal{L}'$ is a reason of $\mathcal{L}$. [7]
- P can play `because` only if the reason given does not contain any arguments already mentioned (i.e., in P's commitment store) but not yet accepted (i.e., not in O's commitment store). We call such arguments *open issues*.
- O addresses the most recent open issue $\mathcal{L}$ (`in`$(A)$ or `out`$(A)$) in the discussion. If O is committed to reasons for $\mathcal{L}$ it must concede $\mathcal{L}$ otherwise O starts to question all reasons that O is not committed to with `why`.
- O can question with `why` only one argument at a time.
- The moves `claim`, `because` and `concede` can be played only if new commitments do not contradict a previous one.
- The discussion terminates when no more moves are possible. If O `conceded` the main argument then P wins, otherwise O wins.

Given the argumentation framework in Figure 2 the interaction between a proponent and an opponent may look as set out in Table 1. Recent research has indicated that it is perfectly possible to use the above-sketched dialogue as a basis for formal argumentation theory. The idea is that an argument is accepted iff it can be defended in rational (structured) discussion. The fully specified theory described in [4] together with the with associated implementation [5] allows

[7] A *reason* for $\{(A, \texttt{in})\}$ is $\{(B_1, \texttt{out}) \ldots (B_n, \texttt{out})\}$ where $B_1 \ldots B_n$ are all the attackers of $A$ in the argumentation framework

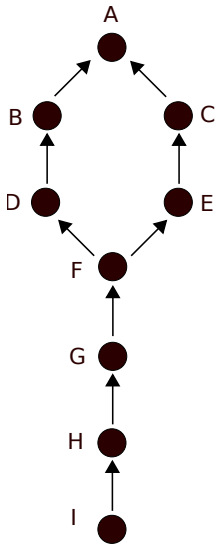


**Figure 2.** An argumentation framework with more than one path. Nodes represent arguments and arrows indicate attack relations. The dialogue in Table 1 shows how this argumentation framework is traversed, and how the status of argument $A$ is determined by observing that it has to be labelled `in` by every complete labelling.

**Table 1.** Table shows the interaction between Proponent and Opponent for the argumentation framework in Figure 2

| | Commitment stores | | | |
|---|---|---|---|---|
| | Proponent | | Opponent | |
| move | in | out | in | out |
| P:`claim` `in` (A) | A | - | - | - |
| O:`why` `in` (A) | A | - | - | - |
| P:`because` `out` (B,C) | A | B,C | - | - |
| O:`why` `out` (B) | A | B,C | - | - |
| P:`because` `in` (D) | A,D | B,C | - | - |
| O:`why` `in` (D) | A,D | B,C | - | - |
| P:`because` `out` (F) | A,D | B,C,F | - | - |
| O:`why` `out` (F) | A,D | B,C,F | - | - |
| P:`because` `in` (G) | A,D,G | B,C,F | - | - |
| O:`why` `in` (G) | A,D,G | B,C,F | - | - |
| P:`because` `out` (H) | A,D,G | B,C,F,H | - | - |
| O:`why` `out` (H) | A,D,G | B,C,F,H | - | - |
| P:`because` `in` (I) | A,D,G,I | B,C,F,H | - | - |
| O:`concede` `in` (I) | A,D,G,I | B,C,F,H | I | - |
| O:`concede` `out` (H) | A,D,G,I | B,C,F,H | I | H |
| O:`concede` `in` (G) | A,D,G,I | B,C,F,H | I,G | H |
| O:`concede` `out` (F) | A,D,G,I | B,C,F,H | I,G | H,F |
| O:`concede` `in` (D) | A,D,G,I | B,C,F,H | I,G,D | H,F |
| O:`concede` `out` (B) | A,D,G,I | B,C,F,H | I,G,D | H,F,B |
| O:`why` `out` (C) | A,D,G,I | B,C,F,H | I,G,D | H,F,B |
| P:`because` `in` (E) | A,D,G,I | B,C,F,H | I,G,D | H,F,B |
| O:`concede` `in`(E) | A,D,G,I,E | B,C,F,H | I,G,D,E | H,F,B |
| O:`concede` `out` (C) | A,D,G,I,E | B,C,F,H | I,G,D,E | H,F,B,C |
| O:`concede` `in`(A) | A,D,G,I,E | B,C,F,H | I,G,D,E,A | H,F,B,C |

participants to discuss whether a particular argument has to be accepted by every reasonable position (complete labelling) that can be taken based on the available information (argumentation framework). The rules of the structured discussion are such that the ability to win the discussion against a maximally sceptical opponent coincides with the argument in question being labelled `in` by each and every complete labelling of the argumentation framework. The structured discussion proposed in [4] is based on Mackenzie-style persuasion dialogue [9, 10], where one can apply moves like `claim`, `why`, `because` and `concede` as described above.

The associated implementation [5] uses a command-line interface, and is written in Python. The argumentation framework can either be loaded from a text file or entered manually. At the highest level, the user has eight commands at his disposal: `question`, `claim`, `load`, `save`, `af_cat`, `af_define`, and `quit`. With `question` the user asks the system about the status of a particular argument (say $A$). The system then responds either with `claim` `in`$(A)$, meaning that $A$ has to be labelled `in` by every complete labelling, with `claim` `out`$(A)$, meaning that $A$ has to be labelled `out` by every complete labelling or with `no commitment` $A$, meaning that neither is the case. In the first two cases, the associated `claim` move is the start of a persuasion dialogue as described in [4], which the user could choose to bypass by immediately conceding the main claim. When the user does a `claim` command, the system responds either by conceding (if it holds the claim that a particular argument has to be labelled `in` or `out` to be correct) or by holding a persuasion dialogue (if the system holds the claim to be incorrect). Although in the latter case, the discussion will in the end always be won by the system (since the ability to win the persuasion dialogue for a particular argument coincides with the argument being labelled `in` by every complete labelling of the argumentation framework [4]) the discussion might still lead the user to valuable insight about why his initial position was wrong. With the `load`, `save`, `af_cat` and `af_define` commands one respectively loads, saves, displays or manually defines an argumentation framework. The dialogue game follows the rules described in [4], with the exception that parties can terminate the dialogue at any point by conceding or withdrawing the main claim.

The source code (GPL) and other necessary files can be downloaded at the project page [8]. The plan is to keep developing it and integrate it with ArguLab [12]. Furthermore, we are currently working on a theory in which arguments are more than just abstract entities, but have an internal structure consisting of a number of reasons that collectively support a particular claim (conclusion). This would result in a richer formalism, and the resulting discussion could be more natural than is the case when (like in the above example) arguments are completely abstract.

## 4 CONCLUSION

In general, the ability to express formal inference as the ability to win a particular type of structured discussion can be helpful for providing explanation to end users about why the system derived a particular outcome. If the user disagrees with the system, then one would essentially do the same as when disagreeing with a colleague: start a discussion.

## ACKNOWLEDGEMENTS

This work has been supported by the National Research Fund, Luxembourg (LAAMIcomp project) and by the Engineering and

[8] `http://code.google.com/p/pyafl/downloads/list`

Physical Sciences Research Council (EPSRC, UK), grant ref. EP/J012084/1 (SAsSy project).

## REFERENCES


[1] M.W.A. Caminada, 'On the issue of reinstatement in argumentation', in *Logics in Artificial Intelligence; 10th European Conference, JELIA 2006*, eds., M. Fischer, W. van der Hoek, B. Konev, and A. Lisitsa, pp. 111–123. Springer, (2006). LNAI 4160.

[2] M.W.A. Caminada and L. Amgoud, 'On the evaluation of argumentation formalisms', *Artificial Intelligence*, **171**(5-6), 286–310, (2007).

[3] M.W.A. Caminada and D.M. Gabbay, 'A logical account of formal argumentation', *Studia Logica*, **93**(2-3), 109–145, (2009). Special issue: new ideas in argumentation theory.

[4] M.W.A. Caminada and M. Podlaszewski, 'Grounded semantics as persuasion dialogue', in *Computational Models of Argument - Proceedings of COMMA 2012*, eds., Bart Verheij, Stefan Szeider, and Stefan Woltran, pp. 478–485, (2012).

[5] M.W.A. Caminada and M. Podlaszewski, 'User-computer persuasion dialogue for grounded semantics', in *Proceedings of BNAIC 2012; The 24th Benelux Conference on Artificial Intelligence*, eds., Jos W.H.M. Uiterwijk, Nico Roos, and Mark H.M. Winands, pp. 343–344, (2012).

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

[7] M. Gelfond and V. Lifschitz, 'The stable model semantics for logic programming', in *Proceedings of the 5th International Conference/Symposium on Logic Programming*, eds., R.A. Kowalski and K. Bowen, pp. 1070–1080. MIT Press, (1988).

[8] M. Gelfond and V. Lifschitz, 'Classical negation in logic programs and disjunctive databases', *New Generation Computing*, **9**(3/4), 365–385, (1991).

[9] J. D. Mackenzie, 'Question-begging in non-cumulative systems', *Journal of Philosophical Logic*, **8**, 117–133, (1979).

[10] J. D. Mackenzie, 'Four dialogue systems', *Studia Logica*, **51**, 567–583, (1990).

[11] S.J. Modhil and H. Prakken, 'A general account of argumentation with preferences', *Artificial Intellligence*, (2013). in press.

[12] M. Podlaszewski, Y. Wu, and M. Caminada, 'An implementation of basic argumentation components', in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pp. 1307–1308, (2011).

[13] J.L. Pollock, 'How to reason defeasibly', *Artificial Intelligence*, **57**, 1–42, (1992).

[14] J.L. Pollock, *Cognitive Carpentry. A Blueprint for How to Build a Person*, MIT Press, Cambridge, MA, 1995.

[15] H. Prakken, 'An abstract framework for argumentation with structured arguments', Technical Report UU-CS-2009-019, Department of Information and Computing Sciences, Utrecht University, (2009).

[16] R. Reiter, 'A logic for default reasoning', *Artificial Intelligence*, **13**, 81–132, (1980).

[17] G.R. Simari and R.P. Loui, 'A mathematical treatment of defeasible reasoning and its implementation', *Artificial Intelligence*, **53**, 125–157, (1992).

[18] A. van Gelder, K.A. Ross, and J.S. Schlipf, 'The well-founded semantics for general logic programs.', *J. ACM*, **38**(3), 620–650, (1991).

[19] G.A.W. Vreeswijk, 'Abstract argumentation systems', *Artificial Intelligence*, **90**, 225–279, (1997).

[20] Y. Wu and M.W.A. Caminada, 'A labelling-based justification status of arguments', *Studies in Logic*, **3**(4), 12–29, (2010).