In M. Baldoni, C. Baroglio, F. Bex, T. D. Bui, F. Grasso & et al. (Eds.),
Principles and Practice of Multi-Agent Systems, LNAI 9935, pp. 3-15. Springer.

A Cognitive Approach to Relevant Argument Generation

Jean-Louis Dessalles

Telecom ParisTech – Université Paris-Saclay
46 rue Barrault, F-75013 Paris, France
www.dessalles.fr – jl@dessalles.fr

Abstract. Acceptable arguments must be logically relevant. This paper describes an attempt to retro-engineer the human argumentative competence. The aim is to produce a minimal cognitive procedure that generates logically relevant arguments at the right time. Such a procedure is proposed as a proof of principle. It relies on a very small number of operations that are systematically performed: logical conflict detection, abduction and negation. Its eventual validation however depends on the quality of the available domain knowledge.

Keywords: Argumentation, relevance, cognition, minimalism.

1 Introduction

Argumentative dialogues constitute the major part of the human language performance. Human beings spend about 6 hours a day in verbal interactions (Mehl & Pennebaker, 2003), uttering 16 000 words on average (Mehl *et al.*, 2007). The two major types of verbal interactions are conversational narratives and argumentative dialogues (Bruner, 1986; Dessalles, 2007).

Argumentative dialogues are produced spontaneously and effortlessly in any group of healthy adult individuals. The ability to generate argumentative moves in spontaneous conversation is crucial to social life, as judgments of rationality and of social competence depend on it. Moreover, various cognitive processes seem to be common to argumentation and to deliberative reasoning (Dessalles, 2008). Modeling the human argumentative ability should therefore be one of the main ambitions in Cognitive Science.

The previous statement relies on the implicit hypothesis that argumentation is a unitary phenomenon. There is no consensus on this. For instance, Walton considers various types of argumentative dialogues as governed by different rules: persuasion, inquiry, negotiation, information-seeking, deliberation and eristic (strife) dialogue (Walton, 1982; Walton & Macagno, 2007). Other authors build on the idea that interacting individuals choose which "dialogue game" they agree to play, among a set of conventional dialogue games available to them (Hulstijn, 2000; Maudet, 2001). The present paper makes the strong assumption that there is a cognitive core that is common to all argumentative dialogues, regardless of the category they fall into.

It may seem quite natural to see argumentative dialogues as a process involving people, as arguments are often described in terms of challenge, commitment, withdrawal or support. As a consequence, the social level is rarely separated from the logical (or knowledge) level. The difference between social aspects and logical aspects is however crucial for the present study. In this paper, we pay attention only to the conceptual and logical structure of argumentation. The challenge is to show that logical relevance can be computed in a phase relying on knowledge and preferences that can be kept separate from other computations based on social roles and social goals. We regard logical relevance as a prerequisite for any other computation regarding argumentation, as no valid argument can be logically irrelevant. The extreme version of our hypothesis consists in saying that the propositional content of argumentative moves follows a definite mechanical procedure, regardless of who makes them. At the knowledge level, it is not possible to tell whether a given argumentative sequence involved three people, two people or was a soliloquy. If this hypothesis is valid, we can study the logical organization of argumentative dialogues while ignoring other issues, independently from their importance in further computations, such as pragmatic goals (convince, influence, gain the upper hand) or saving/losing face. At the knowledge level, it is more urgent to concentrate on the logical relationships between utterances (Quilici *et al.*, 1988). This, of course, amounts to supposing that the logical level has some form of autonomy.

Even if we limit our study of argumentation to computations performed at the logical (or knowledge) level, we must still determine which entities are processed by these computations. In most approaches to argument generation, a pre-computed set of arguments is supposed to be available. Arguments may be propositions that are known for instance to be in relation of support or of attack with another argument (Dung, 1995). Such a set may be given or be computed through a planning module (Amgoud & Prade, 2007). Various questions are then asked, such as finding 'acceptable' arguments (Dung, 1995), or finding best argumentative strategies following rhetorical principles (van Eemeren, Garssen & Meuffels, 2007, 2012). Postulating a graph of pre-existing arguments with attack/support weighted relations may be appropriate in task-oriented dialogues, in which at least some participants are expert not only in the domain of the task, but also in conducting dialogues about the task. Pre-established argument graphs may also be natural to study professional debating behavior, as in political debates. In spontaneous everyday dialogues, however, people are not expected to be experts. They are not even supposed to have any awareness about the possible existence of pre-existing argument collections to choose from. We must assume that every argumentative move is computed on the fly instead of being selected or retrieved. We do not postulate static graphs of arguments; we do not postulate complex procedures such as the search for minimal acyclic paths in such graphs either. It would not be parsimonious to grant such powers to brains. At the other extreme, a purely structural approach that would look exclusively at the surface of the arguments (Rips, 1998) is unlikely to predict the content of utterances.

We choose to settle for the kind of computation considered in BDI approaches, *i.e.* computations about propositions (or predicates), about beliefs and about desires. We must, however, put further restrictions about the kind of computations that can be regarded as cognitively plausible. Since cognitive systems are "embedded systems", we cannot postulate any access to external oracles of truth. We cannot make use of notions such as possible worlds, as long as these "worlds" are supposed to be external entities. And as in any modeling enterprise, we must seek for minimal procedures.

We present here a tentative minimalist model of logically relevant argument generation. Following (Reed & Grasso, 2007), this approach is an attempt of modeling *of* argument (rather that *with* argument). The purpose is to understand the human argumentative *competence* (as opposed to *performance*), rather than using argumentation processes to develop artificial reasoning. In what follows, we will first provide a definition of 'logical relevance'. Then we will introduce the notion of 'necessity', which usefully subsumes attitudes such as beliefs and desires. We then present the conflict-abduction-negation model, before discussing its scope and its limits.

2 Logical relevance

Logical relevance is what makes the difference between acceptable dialogue moves and unacceptable ones, or more generally between rationality and pathological discourse. Logical relevance predicts the conditions in which saying that the carpet is grey is appropriate or, on the contrary, would lead to an expression of incomprehension like "So what?" (Labov, 1997). Note that sentences may be meaningful (*e.g.* "the carpet is grey") and yet be fully logically irrelevant. Philosophical definitions of relevance that rely on the quantity and the cost of inferred information (Sperber & Wilson, 1986) are of little help here, as they are too permissive and do not predict irrelevance. For instance, new knowledge may be easily inferred from "the carpet is grey" (*e.g.* it is not green, it differs from the one in the other room) without conferring any bit of relevance to the sentence in most contexts (*e.g.* during a dialogue about the death of a cousin). Conversely, any sentence can be relevant in an appropriate context (*e.g.* "I asked for a red carpet" or "It doesn't show the dirt"). The point is to discover the kind of logical relationship that an utterance must have with the context to be relevant.

Many task-oriented approaches to argumentation (often implicitly) rely on definitions of relevance or acceptability that refer to *goals*. A move is relevant in these contexts if it helps in achieving one of the speaker's goals. Many spontaneous dialogues, however, occur in the absence of any definite task to be fulfilled. For instance, when people discuss about the recent death of a cousin, they may exchanges arguments about the suddenness or the unexpectedness of the death without trying to achieve anything concrete. Another problem with 'goals' is that there is no way to circumscribe the set from which they would be drawn. Do people who are talking and reasoning about their cousin's sudden death have zero, one, ten or hundreds of goals?

The observation of spontaneous conversation (Dessalles, 1985; 2007) suggests that *problems*, *i.e.* contradictions between beliefs and/or desires, are more basic and more systematic than the existence of goals. For instance, the cousin's sudden deadly stroke contradicts the belief that she was perfectly healthy. The definition of logical relevance that will be used here is straightforward:

A statement is logically relevant
if it is involved in a contradiction
or solves a contradiction.

(for a more precise definition, see (Dessalles, 2013)). It has long been recognized that aspects of argumentation have to do with incompatible beliefs and desires, and with belief revision. “Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires about and what the agent believes” (Bratman 1990). The above definition of logical relevance puts a tight constraint on the kind of move that is admissible in argumentation.

Suppose that the sky is clear and you want to go hiking. Your friend could make a relevant argumentative move by saying “They announce heavy rains this afternoon” because her move creates a contradiction between two desires (hiking and not getting wet). By contrast, saying “They announce heavy rains this afternoon in Kuala Lumpur” would have been irrelevant as long as the argument cannot be related to any contradiction (*e.g.* if you are hiking in the Alps). A further argument from you or your friend such as “I can see some clouds over there” may be argumentatively relevant, for instance by negating one term in the contradiction between ‘observing clear sky’ and ‘having heavy rain soon’. Describing such a move (clouds) as merely ‘strengthening’ the preceding argument (heavy rains) is problematic as long as there is no way to compute such a ‘strengthening’ relation. Fortunately, this is unnecessary: as soon as the intermediary contradiction (clear sky *vs.* rain) is taken into account, the relevance of negating ‘clear sky’ by mentioning clouds becomes evident.

To be logically relevant, people or artificial systems must abide by the constraint above (create or solve a contradiction), or be at risk of being perceived as socially inept. Note that while some models seek for conflict-free arguments (Dung, 1995), we must consider arguments as logically relevant precisely because they create a logical contradiction. Conversely, it is not enough for an argument to be logically consistent with the current state of knowledge. To be logically relevant, an argument that does not create or highlight a contradiction should *restore* logical consistency. In our framework, the only admissible ‘goals’ are prospective situations in which logical consistency is restored (which comes with the strong presupposition that the current situation is regarded as inconsistent for the goal to be considered).

3 Conflicting necessities

Basic attitudes such as *true* and *false* have long been recognized to be insufficient to model argumentation. In line with the BDI approach, we consider that propositional attitudes can be gradual and may include both beliefs and desires. As we experience beliefs and desires as very different mental attitudes, we may expect them to be processed through two radically different mechanisms, one for beliefs and one for desires. We found that, surprisingly, both mechanisms can be naturally merged into a single one[1]. As far as the computation of logical relevance is concerned, the distinction between beliefs and desires can be (momentarily) ignored. To describe the argumenta-

[1] This result was unexpected. Our initial attempts to capture argumentative competence involved separate procedures for epistemic moves (beliefs) and for epithymic moves (desires) (Dessalles, 1985). Gradual simplification in both procedures led them to converge and eventually to merge into a single one.

tion procedure, we use a single notion, called *necessity* (note that the word 'necessity' is close here to the naïve notion and does not refer to a modality). Distinguishing desires from beliefs remains of course essential when it comes to argument wording. The claim is that it plays no role in the computation of logically relevant arguments.

We call *necessity* the intensity with which an aspect of a given situation is believed or wished[2]. Necessities are negative in case of disbelief or avoidance. For the sake of simplicity, we consider that necessity values are only assigned to (possibly negated) instantiated or uninstantiated predicates. We will still use the word 'predicate' to designate them. At each step t of the planning procedure, a function $\nu_t(T)$ is supposed to provide the necessity of any predicate T on demand. The necessity of T may be unknown at time t (we will omit subscripts t to improve readability). We suppose that necessities are consistent with negation: $\nu(\neg T) = -\nu(T)$. The main purpose of considering necessities is that they propagate through logical and causal links.

We say that a predicate is *realized* if it is regarded as being true in the current state of the world. Note that this notion is not supposed to be "objective". Moreover, in the case of contrefactuals, a predicate is realized as long as it is supposed to be true. A predicate T is said to be *conflicting* if $\nu(T) > 0$ when it is not realized, or if $\nu(T) < 0$ in a situation in which it is realized. We say that T creates a *logical conflict* $(T, \nu(T))$ of intensity $|\nu(T)|$. We will consider logical conflicts (T, N) in which N is not necessary equal to $\nu(T)$. Note that logical conflicts (also called *cognitive conflicts*) are internal to agents; they are not supposed to be "objective". More important, *logical conflicts do not oppose individuals*, but beliefs and/or desires. The point of the argumentative procedure is to modify beliefs or to change the state of the world until the current logical conflict is solved. In many situations, solving one logical conflict may create a new one. Argumentation emerges from the repeated application of the argumentative procedure. Within the present framework, argumentative dialogues can be seen as the *trace of a sequential multi-valued logical satisfaction procedure*. One could think of designing a problem solving device that would help people find out optimal plans or beliefs sets. This is not our concern here. The point is rather to discover a procedure that matches human argumentative behavior while remaining cognitively plausible. Our proposal is that such a procedure proceeds by solving logical conflicts sequentially, one at a time.

4 The conflict–abduction–negation model

Our attempts to design a cognitively plausible model of spontaneous argumentation led to a minimal procedure that we describe below. In a nutshell, the procedure tries to detect a logical conflict, and then explores various ways to solve it. Solutions can be found by revising default assumptions, or by revising beliefs and preferences, or by proposing actions that change the state of the world. Our past efforts to design such a procedure involved intricate operations and an external planner. These developments were unsatisfactory due to their cognitive implausibility. To remain cognitively plausible, we stick to the following restrictions.

[2] In (Dessalles, 2008 ; 2013), we show how the notion of complexity (*i.e.* minimal description length) can be used to measure beliefs and desires on a same intensity scale.

R1. Arguments can be potentially any predicate. Their effect on consistency is *computed* (rather than retrieved from pre-stored relations such as 'support' or 'attack').
R2. The knowledge base is addressed by content. We exclude the possibility of scanning the entire knowledge. Queries for rules must have at least one term instantiated.
R3. The procedure is supposed to be sequential, considering one problem (conflict) and one tentative solution at a time.
R4. The procedure should be kept minimal. A cognitive model of natural argument processing cannot consist in a general-purpose theorem prover with the power of a universal Turing machine that derives arguments from complex axioms.

The purpose of R1 and R2 is to avoid making any strong assumption about the nature of the available knowledge. From a cognitive perspective, representing knowledge using rules (as we do in our implementation) is just a commodity, as minds are not supposed to keep thousands of rules expressed in explicit form in their memory (Ghadakpour, 2003). The purpose of R2 is also to make the model scalable. R3 aims at reflecting the reality of human argumentation which, contrary to artificial planning, is bound to be sequential. R4 is not only motivated by cognitive plausibility, but also by scientific parsimony concerns.

Finding a procedure that respects constraints R1-4 while being able to reproduce human performance, even in simple dialogues, proved significantly more challenging than anticipated. After a series of refinements, we succeeded in designing a procedure, named CAN (for Conflict–Abduction–Negation) that we consider for now as minimal (figure 1). This means that we could not find a more concise procedure than CAN that generates only logically relevant arguments and that generates all logically relevant arguments.

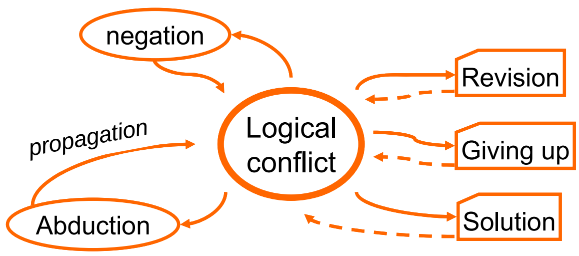

Fig. 1. CAN operations

Figure 2 shows the sequence of operations in CAN. The first step consists in detecting a logical *conflict*. This captures the fact that no argumentation is supposed to occur in the absence of an explicit or underlying logical conflict. The *solution* phase allows actions to be performed and therefore lose there necessity. This phase is where decisions are taken, after weighing pros and cons. For predicates that are not actions, a solution attempt consists in considering them as realized. The next phase is *abduction*. It consists in finding out a cause for a state of affairs. The abduction procedure itself can be considered as external to the model. Necessity propagation occurs in this abduction phase. The effect of *negation* is to present the mirror image of the logical conflict: if (T, N) represents a conflict, so does $(\neg T, -N)$. Note that thanks to negation, positive and negative necessities play symmetrical roles. The *give-up* phase is crucial:

by setting the necessity of T to $-N$, the procedure memorizes the fact that T resisted a mutation of intensity N. Lastly, the *revision* phase consists in reconsidering the necessity of T. This operation represents the fact that when considering a situation anew, people may change the strength of their belief or their desire (the situation may appear not so sure, less desirable or less unpleasant after all).

Conflict:	If there is no current conflict, look for a new conflict (T, N) where T is a recently visited state of affairs.
Solution:	If $N > 0$ and T is *possible* (*i.e.* $\neg T$ is not realized), decide that T is the case (if T is an action, do it or simulate it).
Abduction:	Look for a cause C of T or a reason C for T. If C is *mutable* with intensity N, make $v(C) = N$ and restart from the new conflict (C, N).
Negation:	Restart the procedure with the conflict ($\neg T$, $-N$).
Give up:	Make $v(T) = -N$.
Revision:	Reconsider the value of $v(T)$.

Fig. 2. The sequence of operations in the CAN procedure.

The different phases of the procedure are executed as a "or-else" sequence. This means that if a phase succeeds, the whole procedure starts anew. The solution phase is considered to fail if the intended action fails. The negation phase is executed only once for a given predicate T. The procedure introduces the notion of mutability. C is *mutable* with intensity N if $v(T)$ and N have opposite signs and if $|v(T)| < N$.

Note that the procedure is not following standard deductive reasoning, presumably reflecting the way human beings reason. The abduction phase implements contraposition by propagating negative necessity to the cause. When the propagated necessity is positive, however, the operation is no longer deductive. It can be interpreted as the search for a cause that will be presented as 'necessary'. Since the procedure constitutes a greedy algorithm, each solution is unique by the time it is found, so the distinction between 'necessary' and 'sufficient' gets blurred.

5 Implementing the CAN procedure

The CAN procedure has been designed by successive simplifications of its implementation in Prolog. As suggested by figure 2, the core of the program is now quite short, as we managed to reduce the procedure to a very limited set of operations. Besides this CAN module, the program quite naturally includes two other modules: a domain knowledge and a module named 'world'. For testing purposes, we implemented the domain knowledge as a set of causal and logical rules with default assumptions. We adopted the usual and convenient technique which consists in using the same knowledge to simulate the world (*e.g.* rules are used in forward propagation to update the world when an action has been performed) and to perform abductive reasoning. Moreover, we made the simplifying assumption that the predicates included in the arguments are present in the domain knowledge. These easy options are by no means cognitively plausible. Both the management of the world and the kind of knowledge used for abduction could be implemented in radically different ways (*e.g.* using finer grain representations or even analogue devices) without the CAN procedure being affected.

We tested the model by reconstructing the production of arguments in real conversational excerpts. Below is an example of conversation.

[Context: A is repainting doors. He decided to remove the old paint first, which proves to be a hard work (adapted from French)]

A1- I have to repaint my doors. I've burned off the old paint. It worked OK, but not everywhere. It's really tough work! [...] In the corners, all this, the moldings, it's not feasible!

[...]

B1- You should use a wire brush.

A2- Yes, but that wrecks the wood.

B2- It wrecks the wood...

[pause 5 seconds]

A3- It's crazy! It's more trouble than buying a new door.

B3- Oh, that's why you'd do better just sanding and repainting them.

A4- Yes, but if we are the fifteenth ones to think of that!

B4- Oh, yeah...

A5- There are already three layers of paint.

B5- If the old remaining paint sticks well, you can fill in the peeled spots with filler compound.

A6- Yeah, but the surface won't look great. It'll look like an old door.

If we just keep the argumentative skeleton, we get:

A1- repaint, burn-off, moldings, tough work
B1- wire brush
A2- wood wrecked
A3- tough work
B3- sanding
A5- several layers
B5- filler compound
A6- not nice surface

The challenge is to predict the dynamic unfolding of this argumentative dialogue using a static set of rules representing the domain knowledge. Despite the simplifying assumptions mentioned above, reconstructing the dialogue is a challenging task. The relevant predicates must be selected in the right order and with the right sign (positive or negated) from a (potentially vast) background knowledge base that has ideally been developed independently. For illustrative purposes, we used the following domain knowledge (the sign → stands for causal consequence). Since the program is written in Prolog, it accepts knowledge expressed in first-order logic (*i.e.* with variables). For this simple example, propositions are however sufficient.

```
(C1) burn_off & –wood_wrecked → nice_surface
(C2) filler_compound & –wood_wrecked → nice_surface
(C3) sanding & –several_layers & –wood_wrecked → nice_surface
(C4) burn_off & moldings & –wire_brush → tough_work
(C5) wire_brush & wood_soft → wood_wrecked
(C6) wood_wrecked → –nice_surface
(C7) repaint & nice_surface → nice_doors
```

```
actions([repaint, burn_off, wire_brush,
         sanding,filler_compound]).

default([−wood_soft, −several_layers, −wood_wrecked]).

initial_situation([moldings, -nice_surface, −nice_doors,
                     −wood_soft, −several_layers]).
```

The program needs a few attitudes in addition. These attitudes are represented as numerical values. Only the hierarchy of values is relevant, not the values themselves.

```
desirable(tough_work, −10)
desirable(nice_doors, 20)
```

With this knowledge, the CAN procedure is able to generate exactly the arguments of this conversation excerpt in the right order. The program starts by detecting a *conflict* on 'nice_doors', which is desirable with intensity 20 and yet is not realized. *Abduction* propagates the conflict back to 'repaint' through (C7). 'repaint' is decided as a *solution*, but the conflict is not solved. It is propagated to 'nice_surface' again through (C7), and then to 'burn_off' through (C1). 'burn_off' is performed, but then forward propagation through (C4) generates a new conflict of intensity 10 on 'tough_work'. The trace below illustrates what happens then.

```
 ** Restart.
 Conflict of intensity -10 with  tough_work
 Propagating conflict to cause: -wire_brush
 ------> Decision :  wire_brush
 inferring wood_wrecked from wire_brush
 ** Restart.
 Conflict of intensity 20 with  nice_doors
 Propagating conflict to cause: nice_surface
 Propagating conflict to cause: -wood_wrecked
 Negating -wood_wrecked , considering wood_wrecked
 Propagating conflict to cause: wire_brush
 ------> Decision :  -wire_brush
 inferring tough_work from -wire_brush
 inferring nice_surface from burn_off
 inferring nice_doors from nice_surface
 ** Restart.
 Conflict of intensity -10 with  tough_work
 Negating tough_work , considering -tough_work
 Giving up:  tough_work is stored with necessity 10
 We are about to live with tough_work ( -10 )!
 Do you want to change preference for tough_work ( -10 )?
 ?-  -30.
 ** Restart.
 Conflict of intensity -30 with  tough_work
 Propagating conflict to cause: burn_off
 ------> Decision :  -burn_off
** Restart.
 Conflict of intensity 20 with  nice_doors
 Propagating conflict to cause: nice_surface
 Propagating conflict to cause: sanding
 ------> Decision :  sanding
...
```

The trace shows how the system is able to go back on its decision twice, when it decides that 'wire_brush' and 'burn_off' are bad ideas after all. Note the *give-up* phase in which it keeps a memory of the fact that 'tough_work' resisted a mutating attempt of intensity 10. The *revision* phase is implemented as a question to the user, who sets the preference of 'tough_work' to −30. This triggers a new search for further solutions.

6 Discussion

CAN proceeds by detecting inconsistencies and then by propagating logical conflict to causes. Other systems rely on consistency checking to model argumentation (*e.g.* Thagard, 1989; Pasquier *et al.*, 2006). The present model has several qualities that make it more plausible cognitively.

- Locality: All operations are performed locally or through content addressing. There is no scanning of knowledge.
- Minimalism: The procedure is meant to be the most concise one.
- CAN is recursive, but not centrally recursive. This means that memory requirements do not grow during execution.
- CAN does not loop. The *give-up* phase, by changing necessity values: $v(T) = -N$, prevents abduction from being performed twice identically with the same input. However, repeated revisions may simulate the fact that some human argumentation dialogues go around in circles.
- Despite the fact that CAN ignores people and argument ownership, it captures the dialectical nature of argumentation. Every decision made by CAN represents a move that could be taken on by the same speaker or by another one.

One merit of the CAN procedure is to separate logical relevance processing from the creative part of argumentation. The latter is captured by the abduction procedure. This procedure is external to the model. Thanks to this modularity, CAN may be used as an "argumentative module" in any system that is able to simulate the execution of actions and to perform abduction. The interface between those systems and CAN should involve predicates, but this does not mean that they should use predicates in their internal implementation[3].

At this point, we got little more than a proof of principle. We wanted to prove that part of the human argumentation competence could be plausibly modeled as a fixed procedure. The CAN procedure aims at capturing the rational aspect of argumentation. It does not take any notion of strategy, such as defeating the opponent's counter-arguments, into account. It does not even consider the subjective nature of the social game (convincing game, dispute, counseling…) in which argumentation takes place. However, by enforcing logical relevance, it guarantees the well-formedness of reasoning.

[3] (Dessalles, 2015) shows how predicates can be generated by systems that use perceptual representations.

Conversely, it is hard to imagine how logical relevance could be computed without a procedure like CAN. Even during a quarrel, arguments must be logically relevant, *i.e.* point to inconsistencies or restore consistency. Of course, a general-purpose satisfaction algorithm could produce an optimal solution to restore consistency with minimal attitude change. There is no guarantee, however, that such a solution would be perceived as relevant by human users. People make attempts to restore consistency step by step, following a local procedure like CAN. Logical relevance is checked at each step, changing one attitude at a time. A constraint satisfaction system that would propose a new set of attitudes is likely to be considered irrelevant, as it would be unable to justify the solution stepwise.

Some work remains to be done to turn CAN (or a better version of it) into an operational reasoning module for argumentation systems. Much progress should be made on the abduction procedure, which is currently crudely implemented. The challenge is to find a plausible abduction procedure that would scale up when the size of the knowledge base increases. There are also issues with the accuracy of the available knowledge. This paper shows that the problem of designing argumentative systems can be split in two main tasks: relevance and abduction. Our suggestion is that the CAN procedure captures the relevance part, and that systems based on CAN may produce convincing argumentative dialogues whenever an adequate abduction operation is available.

References

1. Amgoud, L. & Prade, H. (2007). Practical reasoning as a generalized decision making problem. In J. Lang, Y. Lespérance, D. Sadek & N. Maudet (Eds.), *Actes des journées francophones 'Modèles formels de l'interaction' (MFI-07)*, 15-24. Paris: Annales du LAMSADE, Université Paris Dauphine.
2. Bratman, M. E. (1990). What is intention? In P. R. Cohen, J. L. Morgan & M. E. Pollack (Eds.), *Intentions in communication*, 15-32. Cambridge, MA: MIT press.
3. Bruner, J. (1986). *Actual minds, possible worlds.* Cambridge, MA: Harvard University Press.
4. Dessalles, J.-L. (1985). Stratégies naturelles d'acquisition des concepts. *Actes du colloque Cognitiva-85*, 713-719. Paris: CESTA.
5. Dessalles, J-L. (2007). *Why we talk - The evolutionary origins of language*. Oxford: Oxford University Press.
6. Dessalles, J-L. (2008). *La pertinence et ses origines cognitives - Nouvelles théories*. Paris: Hermes-Science Publications.
7. Dessalles, J-L. (2013). Algorithmic simplicity and relevance. In D. L. Dowe (Ed.), *Algorithmic probability and friends - LNAI 7070*, 119-130. Berlin, D: Springer Verlag.
8. Dessalles, J.-L. (2015). From conceptual spaces to predicates. In F. Zenker & P. Gärdenfors (Eds.), *Applications of conceptual spaces: The case for geometric knowledge representation*, 17-31. Dordrecht: Springer.
9. Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, *77*, 321-357.

10. Ghadakpour, L. (2003). *Le système conceptuel, à l'interface entre le langage, le raisonnement et l'espace qualitatif: vers un modèle de représentations éphémères.* Paris: Thèse de doctorat, Ecole Polytechnique.
11. Hulstijn, J. (2000). Dialogue games are recipe for joint action. *4th Workshop on Formal Semantics and Pragmatics of Dialogue (Gotalog-00)*, 99-106. Göteborg: Gothenburg Papers in Computational Linguistics 00-5.
12. Labov, W. (1997). Some further steps in narrative analysis. *Journal of Narrative and Life History*, *7*, 395-415.
13. Maudet, N. (2001). Modéliser l'aspect conventionnel des interactions langagières : la contribution des jeux de dialogue. Thèse de doctorat en informatique, Univ. P. Sabatier, Toulouse.
14. Mehl, M. R. & Pennebaker, J. W. (2003). The sounds of social life: A psychometric analysis of students' daily social environments and natural conversations. *Journal of Personality and Social Psychology*, *84* (4), 857-870.
15. Mehl, M. R., Vazire, S., Ramírez-Esparza, N., Slatcher, R. B. & Pennebaker, J. W. (2007). Are women really more talkative than men? *Science*, *317*, 82.
16. Pasquier, P., Rahwan, I., Dignum, F. P. M. & Sonenberg, L. (2006). Argumentation and persuasion in the cognitive coherence theory. In P. Dunne & T. Bench-Capon (Eds.), *1st international conference on Computational Models of Argumentation (COMMA)*, 223-234. IOS Press.
17. Quilici, A., Dyer, M. G. & Flowers, M. (1988). Recognizing and responding to plan-oriented misconceptions. *Computational Linguistics*, *14* (3), 38-51.
18. Reed, C. & Grasso, F. (2007). Recent advances in computational models of natural argument. *Int. Journal of Intelligent Systems*, *22*, 1-15.
19. Rips, L. J. (1998). Reasoning and conversation. *Psychological Review*, *105* (3), 411-441.
20. Sperber, D. & Wilson, D. (1986). *Relevance: Communication and cognition.* Oxford: Blackwell, ed. 1995.
21. Thagard, P. (1989). Explanatory Coherence. *Behavioral and Brain Sciences*, *12*, 435-502.
22. van Eemeren, F. H., Garssen, B. & Meuffels, B. (2007). The extended pragma-dialectical argumentation theory empirically interpreted. In F. H. van Eemeren, B. Garssen, D. Godden & G. Mitchell (Eds.), *7th Conference of the International Society for the Study of Argumentation*. Amsterdam: Rozenberg / Sic Sat.
23. van Eemeren, F. H., Garssen, B. & Meuffels, B. (2012). Effectiveness through reasonableness - Preliminary steps to pragma-dialectical effectiveness research. *Argumentation*, *26*, 33-53.
24. Walton, D. N. (1982). Topical Relevance in Argumentation. *Pragmatics and beyond*, *3* (8), 1-81.
25. Walton, D. N. & Macagno, F. (2007). Types of dialogue, dialectical relevance, and textual congruity. *Anthropology & Philosophy 8*, 101-119.