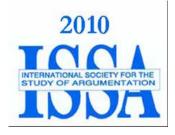# ISSA Proceedings 2010 – Algorithms And Arguments: The Foundational Role Of The ATAI-Question

1. Introduction

Argumentation theory underwent a significant development in the Fifties and Sixties: its revival is usually connected to Perelman's criticism of formal logic and the development of informal logic. Interestingly enough it was during this period that Artificial Intelligence was developed, which defended the following thesis (from now on referred to as the AI-thesis): human reasoning can be emulated by machines. The paper suggests a reconstruction of the opposition between formal and informal logic as a move against a premise of an argument for the AI-thesis, and suggests making a distinction between a broad and a narrow notion of algorithm that might be used to reformulate the question as a foundational problem for argumentation theory.

The paper starts by the analysis of an argument in favor of the AI-thesis (from now on referred to as the AI-argument), distinguishing three premises that support the conclusion (§ 2). We suggest that the interpretation of informal logic as strictly opposed to formal logic might be interestingly analyzed as a move in a strategy to refute the AI-thesis by attacking a premise of the argument: the possibility of expressing arguments by means of algorithms. We are not thereby suggesting that this move was explicitly made by argumentation theorists; nonetheless this counterfactual reconstruction might shed some light on the reasons that opposed argumentation theorists and AI-scholars. In particular, we suggest that the opposition between a formal and an informal approach need not be interpreted only as a way to deal with the peculiarities of ordinary language (analytic philosophy of language answered a similar need without renouncing formal tools, even if only fragments of the natural languages could be formalized), but might also be considered as a way to distinguish the domain of human argumentative rationality from the domain of mechanical computation.

The mentioned strategy will then be compared with other moves directed at the rebuttal of the conclusion of the argument (§ 3). This will allow to distinguish the criticism of the possibility of expressing arguments by means of algorithms from the criticism of the interpretation of Leibniz's logical calculus as the structure of human reasoning, and from the criticism of the thesis that all computable functions can be calculated by a Turing-machine. The comparison of different strategies to rebut the conclusion of the argument will show that a certain understanding of the notion of algorithm is essential in all three strategies: algorithms are considered as computable functions.

We will afterwards discuss a broader notion of algorithm that is often referred to in the literature either as a more intuitive and primitive notion or as a notion that needs to be developed in order to ground recent developments in computation theory and AI (§ 4). We will interpret the narrow notion of algorithm (algorithms are computable functions) as a formal definition that applies only in certain cases but that can fruitfully contribute to an understanding of the intuitive notion.

We will suggest a general characterization of the broad notion as an enlargement of the narrow notion of algorithm. The latter is based on the definitions given by Markov and Knuth (§ 5). Common features of the two notions are finiteness, generality, conclusiveness, while some relevant differences concern the formulation of effectiveness, which needs to be loosened, definiteness, and determinism, which need to be abandoned if one wants to include non-deterministic algorithms, or indefinite algorithms that need to be interpreted by the receiver in a given context, or more generally algorithms that cannot be computed by a Turing-machine.
We will then consider a distinction between a broad and a narrow notion of argument (§ 6), suggesting that, if one interprets formal logic as a sub-domain of informal logic rather than as a radically incompatible research area, then the broad notion of argument can be considered as more primitive and the narrow notion can be seen as a restriction that is useful to understand the nature of arguments but that is also insufficient for certain purposes of argument analysis.

Given this interpretation of the relations between formal and informal logic, several similarities between the broad notions of argument and algorithm are considered (§ 7): not only the history of the relations between a broad and a narrow notion is similar in the two cases, but the two broad notions can be similarly described by difference with respect to the two narrow notions: the

former are informal rather than formal, pragmatic rather than only syntactic, in need of an interpretation rather than unambiguously determined, non-deterministic rather than deterministic. The distinction between a broad and a narrow notion of algorithm will also explain why it was so easy for argumentation theorists to refute the idea that arguments could be expressed by algorithms: they were comparing the broad notion of argument with the narrow notion of algorithm. Once the comparison is made between the two broad notions, certain similarities cannot be ignored, and the fruitfulness of the application of AI to argumentation might be investigated anew.

In the last section of the paper (§ 8) we will go back to the argument sketched out in § 2 in order to claim that the distinction between a broad and a narrow notion of argument, and the developments made by logic, computation theory, AI and argumentation theory in recent years make it easy to rebut the conclusion of the argument. But maybe that is only due to the fact that the idea expressed by it needs to be reformulated in the light of those developments: the question suggested by AI does not concern the emulation of the argumentative reasoning of a single human mind, but rather the emulation of the argumentative practices of several interlocutors interacting with each other in a given context. The question would now be whether a multi-agent system can emulate the interactive reasoning of several human participants in a discussion (from now on referred to as the ATAI-question). This paper does not aim to give a definite answer to the problem, but considers it as a leading idea in the application of AI to argumentation theory and as an open question that is not limited to logic or philosophy of mind but that involves the foundations of argumentation theory itself, and especially its conception of argumentative rationality.

2. The AI-argument and its criticism by argumentation theorists
Between the end of the Fifties and the beginning of the Sixties research into formal logic and AI were oriented by the idea that
(1) human reasoning can be considered as a mechanical computation (Leibniz's calculemus).
The majority of AI scholars also believed in the so-called Church-Turing thesis, which can be roughly formulated as follows:
(2) any computable function can be computed by a Turing-machine.**[i]**
So, if one accepts the further premise that
(3) arguments can be reconstructed as algorithms,

then one can infer by means of (1), (2) and (3) that

(AI-thesis) human argumentative reasoning can be emulated by a machine.

It is well known that a main reason for the revival and development of argumentation theory in the Fifties and Sixties was the reaction to the neopositivist ideas that there could be no rational discussion on judgements of value and that logic could be conceived as a mathematical calculus rather than as a general theory of human reasoning. We would like to suggest that there was a third element of disagreement between argumentation theorists and formal logicians: it concerned the role attributed to algorithms in the representation and understanding of human reasoning.**[ii]**

AI scholars believed that human reasoning was a mechanical computation (1) and thus aimed at restricting the notion of algorithm so as to identify it with a class of computable functions. According to our interpretation the insistence on the opposition between formal and informal arguments could be seen, in the light of recent developments of AI, and independently from the intentions of the argumentation theory scholars that first defended such an opposition, as a move against the AI-argument. Assuming the Church-Turing thesis (2) to be valid, and assuming that arguments can be reduced to algorithms (3), one could derive the conclusion that human reasoning can be emulated by a machine (AI-thesis). But if this is true, there would be no space left for the specific human "rationality" of argumentation. So, while attacking premise (3), one would at the same time rebut the AI-argument, if not attack the AI-thesis altogether. When arguments are defined as classes of sentences of the natural language that could not be adequately translated into any formal language, then they are defined by opposition to algorithms. Besides, it is not uncommon in the argumentation theory tradition to strongly criticize the reduction of arguments to deductive inferential schemes. So, even if we are not suggesting that any argumentation scholar has explicitly advocated this strategy, some of them might agree on the premises of the argument and might be satisfied with its conclusion.

The strategy consisting in the denial of the AI-thesis by refuting premise (3) was useful to distinguish argumentation theory from logic, and thus a condition for the existence of argumentation theory itself, given that if human reasoning does not differ substantially from the reasoning of a machine, there would be no need to distinguish the domain of human rationality from the domain of formal logic (Govier 1987, pp. 204-5).

3. Other strategies to attack the conclusion of the AI-argument
Whether all human reasoning could be emulated by a machine, and whether there was nothing in the human mind that could exceed the powers of a calculating machine became main philosophical questions in logic and philosophy of mind. Among those who tried to refute the AI-thesis there were not only argumentation theorists, but also philosophers and logicians. The move made by argumentation theorists was not the only possible one. Other possible moves included the attack on premise (2), i.e. on the Church-Turing thesis, or on premise (1), i.e. on the mechanical conception of logical reasoning.

Kurt Gödel for example criticized the Church-Turing thesis in a remark on undecidability results, where he reacted to the following version of the thesis: Turing machines can compute any function "calculable by finite means" (Turing 1937, p. 250). There is a huge body of literature discussing the meaning of Gödel's remark although in this paper we will not go into details. What is relevant here is the generally accepted fact that Gödel intended to suggest counterarguments to the idea that the generalized undecidability results might establish bounds for the powers of human reason (Gödel 1986, p. 370). Furthermore, it is relevant that he considered Turing's argument "which is supposed to show that mental procedures cannot go beyond mechanical procedures", as not yet conclusive, because "what Turing disregards completely is the fact that mind, in its use, is not static, but constantly developing, i.e. that we understand abstract terms more and more precisely as we go on using them, and that more and more abstract terms enter the sphere of our understanding. [...] This process, however, today is far from being sufficiently understood to form a well-defined procedure." (Gödel 1972a, p. 306). Even if we admit premise (1), i.e. that human reasoning is a mechanical procedure, its calculations cannot yet be expressed by well-defined procedures.

Another possible strategy to refute the AI-thesis consisted in the attack on premise (1). A similar move had been done already at the end of the 19th century by J. Venn, who argued that even if human reasoning were based on algorithms, it could not be considered as a mechanical computation: "There is, first, the statement of our data in accurate logical language. [...] Then secondly, we have to throw these statements into a form fit for the engine to work with–in this case the reduction of each proposition to its elementary denials. [...] Thirdly, there is the combination or further treatment of our premises after such reduction. Finally,

the results have to be interpreted or read off. This last generally gives rise to much opening for skill and sagacity; [..] I cannot see that any machine can hope to help us except in the third of these steps; so that it seems very doubtful whether any thing of this sort really deserves the name of a logical engine" (Venn 1881, pp. 120-121).

In his 1972 article on the extension of finitary mathematics Gödel interestingly remarked upon a difference between the definition of algorithm occurring in the formulation of Turing's thesis and the intuitive notion of a well-defined procedure or algorithm: the latter is a primitive notion. Although he considers it as adequately expressed by Turing's notion of a mechanically computable function, Gödel adds that "the phrase 'well-defined mathematical procedure' is to be accepted as having a clear meaning without any further explanation." (Gödel 1972, p. 275).

It is interesting to remark that all three strategies are based on the common implicit premise that the conception of algorithms can be adequately described by the notion of computable functions. As Gödel somehow suggested, the notion of algorithm is nonetheless antecedent to Turing's definition and further developments of AI and computation theory have shown that the former might be broader than the latter. In the next section (§ 4) we will thus consider a different understanding of the notion of algorithm that will require a new evaluation of similarities and differences between algorithms and arguments (§ 7). This will also imply that the attack of premise (3) in order to rebut the AI-thesis might not be easily made nowadays.

4. A broad and a narrow notion of algorithm
Recent developments of computation theory and AI suggest that the intuitive notion of algorithm might be broader than the notion of a Turing-machine computable function.
Firstly, there are some procedures that cannot be computed by a Turing-machine. Some of them can nonetheless be computed by other kinds of machines (Gurevich 2000, p. 77 ff.). If an algorithm could be defined as a function that can be computed by a broader class of machines, including the Turing-machine as a particular case, then this notion would be broader than the one given by Turing.

Secondly, there are several notions of a computable function (lambda-computable, general recursive, primitive recursive, partial functions, ...), and there is no

definite evidence that the notion of algorithm should be adequately and uniquely expressed by one of them. As Gödel himself noted in the previously mentioned passages, an intuitive notion of algorithm precedes the notion of a computable function. Blass and Gurevich are even more radical: "it is often assumed that the Church-Turing thesis settled the problem of what an algorithm is. That isn't so. The thesis clarifies the notion of computable function. And there is more, much more to an algorithm than the function it computes. The thesis was a great step toward understanding algorithms, but it did not solve the problem what an algorithm is" (Blass and Gurevich 2003, p. 197).

Thirdly, the definition of algorithm as a computable function was the result of efforts to formulate algorithms that can be computed in a reasonably short time and in a reliable way by machines, but the notion of algorithm historically preceded both the notion of function and the invention of calculating machines. As an example, one could mention the nine chapters on mathematical procedures by Liu Hui written at the beginning of the third century (Chemla 2005, p. 125). Similarly, in the common understanding of algorithms as recipes or procedures to carry out some task (Sipser 2006, p. 142), algorithms are sets of instructions written for human receivers. Unlike Turing-machines, the instructions given to a human receiver need not be completely unambiguous. The context of the algorithm and other pragmatic elements might help the receiver to interpret the instructions of the procedure. So conceived, algorithms might contain procedures that cannot be computed by a Turing-machine.

Finally, the development of multi-agent systems in AI has favoured the investigation of interactive algorithms, that can be implemented on a network of machines: multi-agent systems that can learn from experience and interact in a network. The class of interactive algorithms is so broad as to include randomized algorithms, asynchronous algorithms, and non-deterministic algorithms as well. In other words it includes algorithms that "are not covered by Turing's analysis" (Blass and Gurevich 2003, p. 203).

The analysis of the developments of mathematics, computation theory, and AI shows that a broader notion of algorithm not only preceded the formalized definition given in the 20th century, but has also been the object of research in computation theory. The need for a more precise notion of algorithm induced a narrowing of the notion in order to define it as a Turing-machine computable function. Later on some computation theorists and AI researchers discovered that

this definition might be too narrow to be applied to some interesting examples, and started to progressively broaden the notion of algorithm. We suggest that the narrow notion of algorithm might be conceived as a temporary restriction of a more intuitive and broader notion–a restriction that was particularly useful to understand and formalize certain aspects of the broader notion, but that does not pretend to include all kinds of algorithms.

Rather than broadening the notion of an algorithm by enlarging the class of computable functions or the class of machines to which algorithms correspond – a strategy that has been followed for example by Gurevich – we want to develop here a conceptual analysis of the conditions that the narrow notion usually satisfies and that the broader notion might fail to satisfy. We will claim that a provisionary understanding of the broader notion of algorithm that is at stake in AI and in computation theory could be obtained from the narrow notion if one abandons the conditions of definiteness and determinism, and if one does not formulate effectiveness in a very strict way. If the broader notion of algorithm can be obtained by a modification of the definition of the narrow notion, this does not mean, as we have already suggested in the previous paragraphs, that the narrower notion should be more primitive: on the contrary, the broader notion precedes both historically and conceptually the narrower notion. The latter, though, is easier to formalize, and can thus be used as a starting point for the analysis of the former.

5. A conceptual analysis of the differences
Our suggestion for a characterization of the narrow notion of algorithm is derived, with some modifications and integrations, from the definitions given by Markov and Knuth between the Fifties and the Sixties (Markov 1961 and Knuth 1997). An algorithm is a set of instructions determining a procedure that satisfies the six following conditions: finiteness, generality, conclusiveness, effectiveness, definiteness, and determinism.

Finiteness expresses the fact that the procedure allows, given certain inputs, to reach the goal (decision, computation, problem solving), i.e. provide the desired output in a finite number of steps. Generality guarantees the possibility of starting out with initial data, which may vary within given limits (e.g. certain general classes of inputs are admitted). Conclusiveness expresses the fact that the algorithm is oriented towards some desired result which is indeed obtained in the end if proper initial data are given. Effectiveness requires that the operations

to be performed are sufficiently basic that they can in principle be done exactly and in a finite length of time by the executer (e.g. a man using a paper and a pencil). Definiteness requires that the prescription should be universally comprehensible and precise, leaving no place for arbitrariness. Determinism guarantees that, given a particular input, the procedure will always produce the same output, and will consist in the same sequence of steps.**[iii]**

The mentioned characterization determines a class of definitions of algorithms rather than being itself a definition of algorithm: differences might derive from specific or detailed formulations of each condition. Effectiveness might be for example intended as strongly or weakly polynomial-time complexity; generality might be specified as the requirement that all inputs belong to the class of natural numbers or to the class of real numbers, and so on.

In the light of the brief survey of some occurrences of a broader notion of algorithm given in § 4, we suggest that the broad notion should maintain some features of the narrow notion, allowing other features to be formulated in a more liberal way or abandoned altogether. In particular, the narrow notion of algorithm should be better characterized by finiteness, generality, conclusiveness, and by a 'liberal' formulation of effectiveness. This condition has nonetheless to be at least partially maintained if one wants the algorithm to be concretely computable by some kind of physical machine. The conditions of definiteness and determinism might be abandoned, so as to include non-deterministic algorithms, indefinite algorithms that need some interpretation by the receiver, and algorithms that cannot be computed by a Turing-machine. Abandoning these conditions need not mean of course that all parts of an algorithm would be non-definite or non-deterministic: in order to preserve some kind of effectiveness, considerable portions of the algorithm might have to be definite and deterministic.

6. A narrow and a broad notion of argument
After having introduced a distinction between a narrow and a broad notion of algorithm, we would now like to go back to the definition of argument. This will help a further understanding of premise (3), because in order to discuss if arguments can be expressed as algorithms one should consider which notion of algorithm and which notion of argument is at stake.

In the history of argumentation theory several definitions of an argument have been given. A detailed list of different definitions cannot be presented here, but

two main classes of definitions can be distinguished. The first class contains the definitions of what we will call the narrow notion of an argument, including the Aristotelian scientific syllogisms and formal representations of deductive inferences such as Lorenzen's dialogical moves. Common characteristics of this narrow notion of argument are the formal representation, the central role played by deduction as a core inference, and the context-independent definition of validity. The second class contains several definitions that express a broader notion of argument, including for example the pragmatic conception developed by van Eemeren and Grootendorst (2004), and the informal notion of argumentative schemes developed by Perelman. The definitions that belong to this class are usually informal, context-dependent and based on a diversification of the kinds of relations that can occur between premises and conclusions in order for the argument to be valid: deductive and inductive inferences, but also other schemes, such as analogy or causal relation, are admitted as valid.

The relation between the two classes of definitions can be conceived differently (Johnson & Blair 2002, p. 357 and D'Agostini 2010, p. 35). Some authors consider them as two complementary classes: the informal definition of argument is opposed to the formal notion, as if the two concepts were radically different and applied to different domains (Scriven 1980). Other authors conceive the broad notion as an enlargement of the narrow notion that might be partially or wholly formalized by means of more sophisticated logical tools (non-monotonic logic, dialogue logic, default logic, defeasibility, and so on) (Woods et al. 2002). Following this second interpretation of the relations between the two notions, we have elsewhere argued (Cantù & Testa 2006, pp. 18-21) that the narrow notion might be considered as a temporary restriction of the broader notion that is useful to better understand the notion of inference, rather than as a concept that is radically opposed to it.

In our reading, the opposites informal/formal, syntax/pragmatics, and deductive/non-deductive can be read as relations of subordination rather than as relations of contrariety, and informal logic is considered as an enlargement or liberalization of formal logic. Arguments expressed in the natural language are thus informal not in the sense that they cannot be formal, but rather in the sense that they are "only partially formalizable" by means of the logical tools at our disposal.

The narrow notion is in fact useful to formalize certain arguments that fall under

the broader notion, or at least certain parts of them (Woods & Walton 1982), as well as the formal notion of argument can be used to better understand an argumentation that can never be fully articulated in the natural language, or at least not in the same way.

7. Similarities between arguments and algorithms

Given this interpretation of the relations between formal and informal logic, the history of the relations between the notions of argument is partly similar to the history of the relations between the notions of algorithm. An intuitive broad notion is reduced to a narrow notion in order to be treated formally, but after some time the limitations induced by the narrow notion appear as too restrictive and scholars start considering the possibility of broadening it, even if the broader notion can only be partially formalized or cannot be made as precise as the narrow notion.

The distinction between a narrow and a broad notion that has been presented in the case of algorithms has thus an analogy in the case of arguments. Firstly, the development of argumentation theory, and especially of informal logic as a reaction to the reduction of the notion of argument to logical consequence is similar to the criticism of the reduction of the notion of algorithm to the notion of a computable function. Secondly, several formal definitions of argument were developed in order to make the broader intuitive notion more precise, but after some time they were judged as insufficient to express human reasoning; analogously the notion of a function that is computable by a Turing machine has been recently perceived as too restrictive to express all the possibilities of human computation, although still considered as a good way to make the notion of algorithm precise. Thirdly, as in the case of algorithms, the broad notion precedes the narrow notion both historically and conceptually, even if the latter can be obtained by the definition of the former, if certain conditions are modified or abandoned.

The similarities between algorithms and arguments do not concern only the history of their definitions. If one considers the relation between the two narrow notions of argument and algorithm and the relation between the two broad notions respectively, one might remark certain similarities. The attack made by argumentation theorists on premise (3), i.e. to the claim that arguments can be expressed as algorithms, was based on a comparison of the broader notion of argument with the narrow notion of algorithm. But if one now compares the broad

notion of argument with the broad notion of algorithm, some similarities might need further investigation.

Firstly, the broader notion of argument is not incompatible with a representation by means of diagrams, graphs, procedural forms, and other inferential schemes that can be expressed by algorithms. This is proved by the number of articles and results produced in AI by scholars who developed Toulmin's interpretation of an argument as a procedural form.

Secondly, the attention devoted to pragmatics in argumentation theory is now emerging in computation theory too, especially in the development of algorithms that need to be interpreted by multi-agent systems, whose resources and background knowledge depend on the amount of interaction between the system and the environment and between the agents themselves.

Finally, the interest for the interpretation of the assertions of the interlocutor in the argumentative practice might be fruitfully compared to the interpretation of the information received from an agent in a complex system. The non-deterministic and indefinite aspects of the broader notion of algorithm might usefully be applied to the reconstruction of certain aspects of human argumentative practices.

A deeper investigation of these and maybe other similarities between the broad notion of algorithm and the broad notion of argument might shed some light on a strictly foundational question that will be developed in the next paragraph: are there some specific features of human rationality that explain our argumentative practices and that cannot be reproduced by the mechanical computation of a multi-agent system?

8. Conclusion

Argumentation theory was partly developed in the belief that there is much more to an argument than there is to an algorithm, but the broad notion of argument was compared with the narrow notion of algorithm. Along these lines one could develop a strategy to refute the AI-thesis, i.e. the claim that the argumentative reasoning of the human mind could be emulated by the computation of a machine. But if one considers a broader notion of algorithm, the AI-thesis might be raised anew: is there something in the broader notion of argument that cannot be captured by the broader notion of algorithm?

This question might get a different answer based on recent developments in logic

(non-monotonic logic, default logic, …), in AI (multi-agent systems) and in computation theory (non-deterministic indefinite algorithms). If premise (3) of the argument introduced in § 2 cannot be easily refuted, one might ask oneself if the alternative strategies to refute the conclusion are still viable, after one has abandoned the implicit premise that an algorithm is a Turing-machine computable function.

The claim that the argumentative reasoning of the mind can be emulated by a single machine was mainly a question concerning logic and the philosophy of mind, and not a question concerning argumentation theory, because the reasoning that was at stake there was neither dialectical nor dialogic, but rather a merely monologic calculus. Therefore it is possible to accept premise (3) and still deny the AI-thesis in its original formulation. In the Introduction to *Argumentation in Artificial Intelligence*, J. van Benthem apparently adopts this strategy when he reassures logicians, philosophers and argumentation theorists by saying that no AI theorist believes anymore that machines can emulate humans. Machines are rather useful to improve the understanding of human capacities: "Original visions of AI tended to emphasize hugely uninspiring, if terrifying, goals like machines emulating humans. […] Understanding argumentation means understanding a crucial feature of ourselves, perhaps using machines to improve our performance, helping us humans be better at what we are" (Rahwan and Simari, 2009, p. viii).

This is an easy move, but maybe not too convincing, for even if no AI scholar would claim anymore that a single machine could emulate the reasoning of a single human mind, she could still defend a variant of the AI-thesis reformulated in the light of recent developments of logic, computation theory, artificial intelligence, and argumentation theory:
(ATAI-thesis) a multi-agent system can emulate the interactive reasoning of several human beings.

Recent developments of the applications of AI to argumentation theory suggest that several inter-subjective aspects of human argumentative interactions can be simulated by complex algorithms functioning on systems of interacting machines. It is no longer a question of how far the activities of the brain can be simulated by some physical device, but rather the question is why the application of AI to argumentation theory is so fruitful. For example, there is research on algorithms that produce new arguments, and successful implementations of argument-based

machine learning.

This paper does not aim to give a definite answer to the ATAI-question, but rather to show that the question is still open and cannot be easily liquidated as an obsolete or untenable claim. Once reformulated, the analysis of the ATAI-thesis (i.e. AI-thesis revisited in the light of Argumentation Theory) might have some effects on the foundation of argumentation theory itself, as we will claim in the following, after briefly mentioning what we mean here by foundational questions. According to our understanding, foundational problems in argumentation theory concern the creation of an adequate model that can be used to analyze argumentation practices: according to the reconstruction that we suggested elsewhere (Cantù & Testa 2006), such a foundational role might be played by the notions of dialectics, dialogue, intersubjectivity, pragmatics, but also by some ideal of argumentative rationality. Another relevant foundational issue might concern the bridging of the gap between different traditions (including formal and informal approaches to the reconstruction and evaluation of arguments) in order to provide a general framework for the development of argumentation studies.

Now, the interaction between multi-agent systems is based on communication procedures that have strong similarities with the dialectical and dialogic interactions studied in argumentation theory, inasmuch as it is based on distributive cognition and on pragmatic elements as well as on syntactic and semantic aspects. So, the notions of dialectics, dialogue, intersubjectivity, and pragmatics play a major role also in the applications of artificial intelligence to argumentation theory. The ATAI-question asks if there are grounds for this similarity and implies that, if there are, then one should take the results of artificial intelligence into account when defining such concepts.

Secondly, if mechanical computing can be considered as strictly argumentative, then the relevant features of argumentative rationality might already be captured by the algorithms of a multi-agent system: so, if one wants to claim that human argumentative practices contain some specificity ("the" rationality of argumentation), then one should exhibit some features (other than pragmatics and interaction) that could not be captured by the activity of some multi-agent system, and this, we believe, is a foundational task.

Thirdly, the ATAI-thesis in connection with the distinction we suggested between a broad and a narrow notion of algorithm might suggest a new and fruitful way to

bridge the gap between formal and informal approaches to argumentation theories, providing a new framework that could include both without misrepresenting their differences and peculiarities.

NOTES

i The notion of a Turing-machine was first introduced by Alan Turing in 1937 in order to analyze the notion of computability. It is an ideal state machine made of an infinite one-dimensional tape divided into cells, each one able to contain one symbol, either '0' or '1'. The machine has a read-write head, which scans a single cell on the tape at a time, moving left and right along the tape to scan successive cells. The machine actions are completely determined by the initial state of the machine, the symbols scanned by the head in the cells, and a list of instructions of the kind "if the machine is in the Initial State S_0 and the current cell contains the Symbol y, then move into the Next State S_1 taking Action z".

ii Cf. for example Toulmin 2001, p. 96, where the search for algorithms is criticized as a correlate of the search to ground objectivity in a unique methodological standpoint: "These arguments may leave mathematically-minded readers with a sense of loss. The dream of formal "algorithms" for guiding scientific procedures has a charm that will not quickly dissipate. For those who value mathematical exactitude above all other kinds of precision as the model for scientific inquiry, the alternative message of "different methods for different topics" will be a disappointment. Yet, over the centuries, we have been obliged to recognize a spectrum of different kinds of methods (in the plural) for sciences ranging from Newton's Planetary Theory—strictly factual and value-free, and in a style close to that of Euclid's Geometry—by way of empirical or functional sciences like geology, chemistry, physiology, and organic evolution, to those human sciences in which attempts to maintain value-neutrality finally proved vain."

iii The notion of conclusiveness, taken from Markov 1961, is similar to the notion of determinism, but might be fruitfully distinguished from the latter if one accepts Gurevich's characterization of non-deterministic algorithms as a special class of interactive arguments. "Imagine that you execute a non-deterministic algorithm A. In a given state, you may haperlve several alternatives for your action and you have to choose one of the available alternatives. The program of A tells you to make a choice but gives no instructions how to make the choice. [...] Whatever you do, you bring something external to the algorithm. In other words, it is the active environment that makes the choices." (Gurevich 2000, p. 25.) Gurevich's

algorithm might be conclusive, because once the choice is made, the desired output might indeed be obtained, but it is non-deterministic, because depending on the choice there might be more than one sequence of steps leading from the input to the output. Besides, the algorithm might still be definite, at least in the sense that the arbitrariness does not depend on an ambiguous formulation of the algorithm, which allows for different interpretations, but rather on the introduction in the algorithm of something external to it.

REFERENCES

Blass, A., & Gurevich, Y. (2003). Algorithms: A quest for absolute definitions. *Bulletin of European Association for Theoretical Computer Science*, 81, 195-225.

Cantù, P., & Testa I. (2006). *Teorie dell'argomentazione* [Theories of Argumentation], Milano: Bruno Mondadori.

Chemla, K. (2005). The interplay between proof and algorithm in 3rd century China: The operation as prescription of computation and the operation as argument. In P. Mancosu, K. F. Jorgensen & S. A. Pedersen (Eds.), *Visualization, Explanation and Reasoning Styles in Mathematics* (pp. 123–145). Dordrecht: Springer.

D'Agostini, F. (2010). *Verità avvelenata. Buoni e cattivi argomenti nel dibattito pubblico* [Poisoned truth. Good and bad arguments in the public debate] .Torino: Bollati Boringhieri.

Eemeren, F.H. van, & Grootendorst, R. (2004). *A Systematic Theory of Argumentation. The Pragma-Dialectical Approach*. Cambridge: Cambridge University Press.

Gabbay, D.M., Johnson, R.H., Ohlbach, H.-J., & Woods, J. (Eds.) (2002). *Handbook of the Logic of Argument and Inference. The Turn Towards the Practical*. Amsterdam: Elsevier.

Gödel, K. (1990). *Kurt Gödel: Collected Works. Vol. II: Publications 1938-1974* (S. Feferman, J. W. Dawson , S. C. Kleene, G. H. Moore, R. M. Solovay & J. van Heijenoort, Eds.). New York: The Clarendon Press.

Govier, T. (1987). *Problems in Argument Analysis and Evaluation*. Dordrecht: Foris.

Gurevich, Y. (2000). Sequential abstract state machines capture sequential algorithms. *ACM Transactions on Computational Logic*, 1, 77–111.

Johnson, R.H., & Blair, J.A. (2002). Informal logic and the reconfiguration of logic. In D.M. Gabbay, R.H. Johnson, H.-J. Ohlbach & J. Woods (Eds.), *Handbook of the Logic of Argument and Inference. The Turn Towards the Practical* (pp. 339-396),

Amsterdam: Elsevier.

Knuth, D.E. (1997). *The Art of Computer Programming*. Vol. 1. Reading, MA: Addison-Wesley.

Lorenzen, P., & Lorenz, K. (1978). *Dialogische Logik*. Darmstadt: Wissenschaftliche Buchgesellschaft.

Markov, A.A. (1961). *Theory of Algorithms*. Jerusalem: The Israel Program for Scientific Translations.

Perelman, C., & Olbrechts-Tyteca, L. (1958). *Traité de l'argumentation. La nouvelle rhétorique*. Paris: Presses Universitaires de France.

Rahwan I., & Simari, G.R. (2009). *Argumentation in Artificial Intelligence*. Dordrecht: Springer.

Scriven, M. (1980). The philosophical significance of informal logic. In J.A.Blair & R.H. Johnson (Eds.), *Informal Logic: The First International Symposium* (pp. 148–160). Inverness, CA: Edgepress.

Sipser, M. (2006). *Introduction to the Theory of Computation*. Boston, MA: PWS.

Toulmin, S. (2001). *Return to Reason*. Harvard: Harvard University Press.

Turing, A.M. (1937). Computability and λ-Definability. *The Journal of Symbolic Logic*, 2: 153–163.

Venn, J. (1881). *Symbolic Logic*. London: Macmillan & Co.

Woods, J., Johnson, R.H., Gabbay, D., & Ohlbach, H.J. (2002). Logic and the practical turn. In D.M. Gabbay, R.H. Johnson, H.-J. Ohlbach & J. Woods (Eds.), *Handbook of the Logic of Argument and Inference. The Turn Towards the Practical* (pp. 1-40). Amsterdam: Elsevier.

Woods, J. & Walton, D.N. (1982). *Argument. The Logic of Fallacies*. Toronto: Mc Graw-Hill.