

# ALGORITHMS IN COGNITION, INFORMATICS AND LOGIC

## A Position Manifesto

by

D. Gabbay and J. Siekmann

The traditional view of an algorithm  $A$  is that it is a recipe, a sequence of exact steps designed to facilitate the execution of some goal involving, say, an entity  $E$ .

This view goes back to the Greeks with such well known examples as Euclid's algorithm and can be found even earlier in the Babylonian times. It is the view that has been taken up ever since by mathematicians and more recently in computer science: myriads of such algorithms are known and recorded. General investigations tried to make these notions precise and the last century in particular turned out to be very fruitful indeed with the foundational work in mathematics and computer science.

So, for example if  $E$  is a list and we wish to order it according to some measure, we may have a rich choice of algorithms for achieving this goal. They may differ in style and efficiency. Taking this view of an algorithm  $A$ , two properties come immediately to mind.

1. It is important how time and resource efficient the algorithm is.
2. The algorithm is not part of the declarative intrinsic nature of the entity it serves. Put differently – whatever the nature of the list is – the algorithm for ordering it is external to that nature.

This point of view of the relationship between algorithms and the entities they serve is commonplace. The research communities concentrated their efforts on finding algorithms for various tasks, classifying and optimizing them as well as developing general theories (of algorithms) for comparing them. But, taking a platonic view, algorithms are not full entities in the ideal platonic heaven and certainly, to whatever extent that they do reside there, they are not intrinsically related to the entities they handle.

We put forward to you the reader a different view. We claim that algorithms have existence and nature beyond the local goals of what they do and that in general algorithms governing and serving an entity form an integral part of the nature of that entity.

An example taken from some century old craftsmanship is say the delineation of an elaborated cupboard in the 17<sup>th</sup> century made by a technical draughtsman to be given to the master joiner: the expert will see the drawing probably just as any layman may, but immediately perceives and mentally visualizes the algorithms (or workflow to use more modern terminology) required to build this cupboard. The declarative representation is the same, but these algorithms distinguish the master from the apprentice or the journeyman – as the second author of this manifesto learned from the century old German "Handwerkskammer and Tischlerinnung", when he started off from home as a joiners apprentice.

In particular the efficiency and "Gründlichkeit" (thoroughness) by which these algorithms are executed will show in the final products and a well trained expert will be able to spot the difference in a split second.

Starting from the general debate in the mid nineteen seventies in artificial intelligence on procedural versus declarative representation, which found its way into many technical fields such as logic programming or object oriented programming, we take the view that an entity is given by two parts: its declarative representation, i.e. its description, and the algorithms associated with it.

Put in a form of a slogan, we say

Object = Properties + Algorithms.

This slogan is similar to

Matter = Elementary Particles + Binding Energy.

Furthermore, we put forward that two algorithms can have distinct declarative flavors and are not just combinatorial “local hacks” and such flavors have significance across different application areas in which they are used. In fact the intuitive human mind can recognize such flavors and see affinity between algorithms in different areas which may differ in details but are similar in flavor and approach.

An animal can be described by its properties such as color, weight and height and its parts, but just as importantly by its pattern of behavior that make it distinct. A manufactured object has certain properties, but more likely than not, it also computes some algorithm that makes it useful: our wristwatch computes the time mechanically and an iPod uses MP3 and other electronic computations. The first radio or the most recent integrated television and web technology all derive their most interesting features from the way they function, i.e. how they compute some algorithms implemented in software or represented directly in their hardware.

To go back to our more mundane and technical example of ordered lists, a list E is characterized not only by the nature of its ordering (by size, precedence, by importance, etc.) but also by the algorithm A serving it: how to restore the order should a new element be landed in the middle of the list or should its order be disrupted and needing to be restored. So our list E is not just “E” but  $\langle E, A \rangle$ . And we should perceive the same E with different ordering algorithms as “different lists”.

The first author of this manifesto hit upon this idea some twenty years ago in connection with logic. He was looking at goal directed formulations of various classical and non-classical logics and noticed that different logics are obtained by slight variations in the algorithm. Thus one can associate declarative properties (logic axioms) to what seems to be purely algorithmic moves (e.g. how you do your garbage collection may determine what logic you are in).

Indeed we humans have an independent perception of a proof theoretical method (e.g. tableaux, resolution, truth values, etc.) as compared with our perception of the logic itself as a declarative entity. The above prompted the first author to declare that the nature of a logic itself cannot be separated from its algorithmic presentation, thus classical logic presented as a Gentzen system is not the same logic, from the point of view of applications, as classical logic presented as a resolution system (see: D. Gabbay and N. Olivetti: Goal directed proof theory, Kluwer, 2000).

Furthermore there may be a trade-off between algorithmic optimisation constraints on a proof procedure for a logic L, and the declarative strength of L. In some cases we get a different, weaker but known logic as a result of applying these constraints, and in many cases their effects are not known. For example the connection graph method, in the theory of resolution for classical predicate logic, is a striking example of a combination of purely declarative (resolution) rules and algorithmic control (of resolution through the connections). We know that classical logic is complete for the resolution rules. However we do not know whether there is correctness and completeness for the system where resolution is controlled by a connection graph. For more than a quarter of a century this problem has been open now and it turned out to be one of the major open problems in our field (see: J. Siekmann, G. Wrightson, Strong Completeness of Kowalski’s Connection Graph Proof Procedure, Springer Lecture Notes on AI, vol. 2408, p. 231, 2002).

The appropriate way to look at such a system is as a mixture of a declarative component presented by the complete proof procedure and the algorithmic restrictions on it. The combination of the two is a mixed presentation of a new system posing this kind of challenge.

This mixed presentation is present everywhere: in applied logic, AI, linguistics, computer science and even philosophy. We give more examples:

- Consider a representation of an agent (irrespective of the exact theory used). Any agent will have beliefs, belief revision, reasoning mechanisms (such as abduction) and more. Some of these features are declarative, others are algorithmic. The agent's nature is determined by both, i.e. these algorithms as well as the declarative content build the “personality” of the agent.
- Another example is from robotics. Fiora Pirri visited us some time ago telling us she won the robot rescue competition because her robot was the only one able to recognise injured persons in a room (together with many other uninjured bodies). We were impressed by this because her method was not declarative but algorithmic: the robot views a person as a compilation of body parts according to a certain algorithm. If the candidate injured person does not 'compile' correctly from its body parts, then it is an anomaly and the robot concludes that the person is indeed injured, i.e. an object is characterised not only by its declarative properties but also by the algorithms relating to its parts and the algorithms relating to how it is to be used or interact with other objects.
- This view, i.e. the inseparability of the two aspects, the algorithmic and declarative nature is of course the essence of logic programming and the above robot example is more generally present in standard object oriented programming.
- Another example can be found in theories of ambiguity and parsing in natural language. The same string of words constructed in two different ways can mean two different things. The analysis of entities such as pronouns and quantifiers require an algorithmic approach over the syntax (as evidenced by theories like Discourse Representation Theory and Dynamic Syntax). In many cases the reference of a pronoun is not determined by the grammatical algorithms alone. To identify the appropriate linguistic interpretation we use common sense and non-monotonic reasoning on the context taking into account factors such as relevance and computational effort.
- Further well known examples are SAT-procedures and model checking: essentially they are logical, i.e. declarative in nature, but these two fields derive their importance from the outmost engineering capabilities in algorithmic design and implementation.

There is also a common sense human perception of the algorithmic nature of entities. Consider for example a very strict catholic cardinal  $C_1$  and a strong Islamic Ayatollah  $A_1$ . They are certainly opposing entities in their nature and opinions. Compare them with more tolerant understanding and behaviourally different colleagues  $C_2$  and  $A_2$ . The common sense reasoner may perceive that  $(C_1, A_1)$  and  $(C_2, A_2)$  have more in common than  $(C_1, C_2)$ ,  $(A_1, A_2)$ . This means that we are more perceptive of the algorithmic behavioural aspect of the entities than their declarative aspect.

The hardcore algorithmic man may remain unconvinced by the above argument. He may challenge us asking “what is the role of complexity and efficiency” in this “declarative” view of an algorithm? Given an entity  $E$  with algorithm  $A$  associated with it, suppose we manage to make  $A$  more efficient by slightly improving upon it. So we now have  $(E, A')$  instead of  $(E, A)$ . We would not want to say that these are different entities but on the other hand we may not be able to say that the change from  $A$  to  $A'$  is not significant. It may be an extremely significant improvement. So what do we have to say about this?

Our answer lies in the aspect of potential use and potential interaction. So the two entities (E, A) and (E, A') are not the same because they have different potential uses and roles to play within a larger system.

Take for example the superstar Claudia Schiffer and her algorithm being her mode of behaviour in public glamorous life. Let us make a small change in her behaviour, declaring that she converted to faith and religiousness and will never look at another man, except her husband. Such a small algorithmic change may make a huge difference to her social interactive life to the extent that she becomes perceptively a different person.

Potential behaviour is central to everyday life. We make our decisions today based on potential behaviour of our human and natural environment tomorrow and so a small change in the same algorithm governing some aspect of our environment may change the potentialities involved and therefore affect our decision. So (E, A) and (E, A') may be seriously different entities if they have different potentialities on account of A' being more efficient.

So what we are eluding to here is the idea that algorithms are part of the treasure trove of human knowledge collected over millennia, just like the heritage in art, social ideas (such as democracy) and any other cultural and scientific heritage we maintain. This view includes technical algorithms such as Euclid's well known case just as much as say "First-Come-First-Serve" which forms part of the British way of life and socio-cultural identity to be witnessed in any London bus queue.

One of the first general repositories of this nature was set up and developed over a time span of more than two decades (see: K. Mehlhorn, S. Näher, LEDA, A Platform for combinatorial and geometric Computing, Cambridge University Press, 1999). It is now maintained by the German Max Planck Institute and a small company. Another example is the CGAL ) repository (Computational Geometry Algorithms Library).

An essential aspect of any algorithm is its logical nature and this is seen as its essence by many researchers, who recognise logic as being the foundational science.

Algorithm = Logic + Control

This is a slight variation of the well known slogan of R. Kowalski and the logic programming community. However this may be, i.e. runtime behaviour may be more than just "control", the essential foundational studies of the last century in logic and recursion theory provided the clarification of the nature of algorithms we enjoy today and not only because of his, we see this journal as a sister to the Journal of applied Logic (JAL)

The first author has an image about logic and algorithms which he repeatedly tells his students: when God created the world he sprinkled around a little bit of logics (and procedures) to act as spice and bonding for his creation. Logics and algorithms are everywhere. It is the job of the research community to figure out what was given to us. We already have the journal of applied logic as the brother to serve us and now we want to establish its sister journal JoA. And of course we have another brother, the Journal of Artificial Intelligence (JAI).

There is experimental support for the claim that objects are perceived together with the algorithms governing them. We refer to the work of professor Giacomo Rizzolatti and his colleagues. (See for example: Lacoboni M, Molnar-Szakacs I, Gallese V, Buccini G, Mazziotta JC et al. "Grasping the intentions of others with one's own mirror neuron system"(2005) Plos Biol3(3) :e79

The experiment, schematically described in our own words, is more or less as follows: We show a subject, (man or a trained monkey) two cups. One with a handle suitable for drinking from, and one with a different handle, not suitable for drinking. The subject responds to the drinking cup by increased activity in two regions of the brain, one dedicated to motoric actions and one known to be dedicated to the recognition of intentions and goals

The response to the non drinking cup activates only the motoric region.

This experiment and others like it clearly show that part of the perception of the drinking cup is the algorithm associated with it which includes the drinking intention. The algorithm associated with the non-drinking cup which has a different handle, does not include the drinking intention.

Successful research by the community requires successful social organisation. There are two main demands we like to serve:

- (A) A general forum to present algorithms from as many areas as we shall encounter in the coming decades to foster cross fertilization between these by and large unrelated areas.
- (B) To establish a repository of algorithms and the appropriate general infrastructure to maintain it.

(A) There are many cases of research paper submissions that are considered out of scope of current area-specific journals. They may be considered too algorithmic for one area, and not algorithmic enough or relevant for another. We need a new journal which can provide a natural home for many such papers of sufficient academic merit. Such a journal will also be able to welcome a variety of papers with relevant algorithmic contents which would have otherwise been scattered in other journals of specialised communities and viewed locally as not mainstream. Here are some examples:

- A paper on complexity of some modal logics is likely to be published in a pure mathematical logic journal. The author, most likely a mathematician, can now consider submitting it to this journal hoping it will get the attention of users not only in modal logic. Take the now famous similarity between modal logic and description logics in the semantic web.
- A researcher with an algorithmic paper about how an agent is to adjust to a changing environment is likely to publish her paper in an autonomous agent journal. However, she may now consider publishing it in the agent section of this journal, realising that readers from the belief revision section or even the database section may be interested.
- Papers on automated reasoning with algorithms of interest to description logics or relational databases will have an alternative home to the automated reasoning journals.

We believe the very idea of such a general forum (in areas including pure logic, AI, automated deduction, social decision theory and the philosophy of science) is a novelty in itself. Today's journals for algorithms are mostly journals for mathematical algorithms and for that reason have not always managed to capture the corresponding (applied) scientific community. As a consequence, they are not doing as well as they could.

We believe that authors of papers with algorithmic content, who believe their paper has interest beyond their own application area, may prefer to submit to this new general forum

(B) Just as the world wide web increasingly represents a large part of our human knowledge and begins to supersede its predecessor storage devices: - for example books – we hope that such a repository will store eventually most known algorithms and their description in a www-manner. Moreover such a repository will provide an infrastructure and execution environment, so algorithms can be searched, checked, downloaded and used. This is in common with the LEDA repository of Kurt Mehlhorn and the repository CGAL (the repository of algorithms in computational geometry).

Proprietary rights will be respected and served appropriately, but we expect by far the largest proportion of these algorithms to be public domain.

So the journal itself will encourage at least two different submissions: the usual scientific paper with algorithmic content and secondly a possibly short description of an algorithm to be stored in the repository. As our managing director Sweitze Roffel from Elsevier assured us: “even if each issue of this journal is as thick as a telephone book”, they will go along with us and publish.

This new journal is strongly supported by the International Federation of Computational Logic. The Federation took over an existing discontinued journal of Elsevier and converted it to this new journal to be perceived as a sister journal to its brother JAL. The Federation is a public registered charity, its operations and accounting is scrutinised and regulated by the Charity Commission in London and it is dedicated to the promotion of computational logic in the international community.

We are grateful to Elsevier allowing us to take over and convert this journal and appreciate their support of the federation and its goals, thus demonstrating also their general commitment to the scientific communities involved.

To summarise, we put forward to the community to recognise that algorithms are part of the nature of entities and recommend that the communities become organised socially to adjust and support this view.

Version 19.08.2007-10.30Uhr

Expanded 24.10.07

Expanded and corrected: 22.1.2008 and again 23.1.2008