

Argument from AI summary: How does asemantic computing differ from traditional distributed computing?

Marius Buliga

Institute of Mathematics, Romanian Academy
P.O. BOX 1-764, RO 014700
București, Romania

Marius.Buliga@imar.ro , mbuliga@protonmail.ch

23.10.2023

Abstract

We explain what is an argument from AI summary [7] and what is asemantic computing [1]. As an example of such an argument from AI summary we provide the answer given for the question in the title by the LLM from [5]. Then we give a real life example about how to counter a valid argument by AI summary.

1 Introduction

An argument from AI summary [7] aims to establish if the written communication of an idea is defective or not, by using an AI tool as a mirror which reflects back the message sent by the human author of the idea. In order to construct such an argument the human author asks an AI to provide a summary of the thesis shared by the author. The summary is then checked by the author. In the case when the author confirms the summary, then the conversation with the AI can be used as proof that the communication of the idea is conformal with the intentions of the author.

With such a proof, the author may argue that if (even) the AI gets the intended message, then probably the message is clear enough for other human readers.

Mind that an argument from AI summary does not suppose that there is an a priori correct meaning of the message, nor that the AI tool has any understanding of the message. The human author inputs the message into the AI tool then verifies that the output reflects the author's intentions. In the case when the author does not think that the AI tool output is conformal with the author's

intentions, then such an argument from AI summary indicates that the AI tool is not up to the task or that the author should work more on the message.

In the following is given the AI [5] answer to the question from the title and as proof the archived answer [6]. As the author of [1], I confirm that the AI answer uses my article in a way which corresponds well with what I intended to communicate in [1].

2 Asemantic computing

As explained in Molecular computers which are based on graph rewriting systems like chemlambda, chemSKI or Interaction Combinators, the repository which contains also the article [1], graph rewriting systems are a very promising direction for building decentralized, distributed computing systems, as well as an inspiration for real life molecular computing. Among such graph rewriting systems we list: Lafont Interaction Combinators [2], chemlambda [3], chemSKI [4]. Computing with such systems is local in space and time and therefore there is no need for a global semantics, from the point of view of computation.

Classically there is a 3 stages process which uses graph rewriting. We want to solve a problem, therefore we start from a program, or more generally from a particular class of term rewrite systems (such as for example lambda calculus, but not combinatory logic).

Meaning to structure: The program is then compiled to a graph. This can also be seen as a language to structure problem. The language can be a term rewrite system, the program can be a term, the structure can be an abstract syntax tree. Many other examples exist, which can be put into the form of transforming something which has a meaning (for a human), to something which can be processed by a machine (in this case a graph).

Structure to structure: transform the initial graph into a final graph, by using the graph rewriting system and an algorithm for the choice and order of application of the rewrites. Usually this problem is complicated because we want to have a good translation not only of the meaning to structure, but also of the term reduction into graph reduction. For example, when we write a program we expect it to "do" something, and we would like that the execution of the program by the decentralized machine (ie the structure to structure part) to be a translation of what we imagine the program is doing. As an example, if our program is a lambda term, which is compiled into a graph, we would like the graph rewriting to be compatible with the term rewriting, ie the beta reduction term rewriting rule. These semantic constraints which are put in the use of graph rewriting are incompatible with decentralized computing.

Structure to meaning: transform the final graph into the language, in order to obtain the answer of the problem. For example, if the initial program was a lambda term, we compile it into a graph, then we reduce the graph, then we

translate back the final graph into a lambda term. In other examples we just want to translate the final graph into something which has a meaning for us, into the solution of the problem.

In all such applications of graph rewriting, we position at the level of meaning, we descend at the level of structure, where we use graph rewriting as a tool and then we translate back the result into a meaningful output.

In contrast with the classical approach, it is argued that distributed computation with graph rewrite systems should skip the initial and final global semantics conversion. Instead, we should regard as the natural level of computation the one called "structure to structure". The design of the initial structure (if any, or if it makes sense to think about an initial global graph) could use the step called "meaning to structure" only as a tool which serves the structure to structure goal. Likewise, the extraction of meaning from the main computation part can be only based on local in space and time information. Finally, the graph rewriting algorithm should be as well local in space and time. Such a computation style which sits mainly at the local graph rewrite level is called "asemantic computing" in [1].

3 The answer to the question

The conversation with the AI was started by the demand to summarize [1]. Then the AI suggested the question from the title. In the following is reproduced the answer to this question. The whole conversation is archived at [6].

The edits of the answer are only the replacement of the text "geeksforgeeks.org" with [1] for the links provided in the AI answer (only the links name are changed, all links point to [1]).

How does asemantic computing differ from traditional distributed computing?

Answer | Phind V6 Model

Asemantic computing differs from traditional distributed computing in several key ways:

1. Global Control and Meaning: Traditional distributed computing systems require a global control and meaning. This is because they are designed to perform tasks that require a global understanding of the system's state. In contrast, asemantic computing systems do not require a global control or meaning. They operate locally in time and space, without needing to understand the overall state of the system [1].

2. Semantics: Asemantic computing does not have semantics, which are the meanings of symbols in a system. This is a significant difference from traditional distributed computing systems, which rely on semantics to understand and interpret the data. In asemantic computing, the data is processed without any inherent meaning [1].

3. Turing Machines and Graph Rewrite Systems: Asemantic computing uses Turing machines and graph rewrite systems, which are local and do not require global control or meaning. In contrast, traditional distributed computing systems often use term rewrite systems, which are non-local and require global control [1].

4. Confluence: Asemantic computing does not require confluence, which is the property of a graph rewrite system where if we have a graph that can be rewritten into one without any further possible rewrites, then this final state is unique. This is undesirable for life-like distributed computations, where final states are to be interpreted as death and they have to be recycled somehow by another mechanism [1].

5. Extracting Meaning: The meaning we can extract from an asemantic computing system is only local. In traditional distributed computing systems, we can extract global measures of agreement and synchronization. However, in asemantic computing, we cannot extract precise global measures of agreement or synchronization [1].

6. Programming and Usage: Asemantic computing systems are unclear how to program or use them in the old ways. However, they can be used as if they are a living ecosystem, an extension of the meatspace [1].

4 How to counter a valid argument by AI summary

The article would not be complete without this section. Hopefully, after the initial version of the article was communicated, a real life example of a human way to counter an argument by AI summary was obtained.

In order to understand the how the argument can be attacked, let's analyze how the argument is built, in time.

Step 1. The author has or claims a thesis, in the form of a document A.

Step 2. Then the author uses an AI tool as a black box. The author inputs A into the black box, which responds (outputs) the summary B.

Step 3. Then the author of A judges B and arrives to the conclusion C that B shows that A is clear enough.

As a part of a discussion, the author then shares A, B, C. For this article A is [1], B is [6] and C is the statement from the end of the Introduction section of this article.

Therefore, the structure used involves a human (the author), a black box (the AI tool), a judge (the human) and evidence of the process (A, B, C). It is important that the black box is available to all the participants in the discussion.

In order to counter the author argument, the receiver uses the same structure, in a reversed time order. Suppose that the receiver poses as an institution (therefore not a human). The receiver has access to it's own black box (a part of the institution itself), but differently from the AI tool, only the receiver can access this black box.

Step 3'. The receiver takes the author, together with A, B, C and classifies all of them as evidence (say request C') for a process of admission of the argument. Differently than the author process, which was completely open, the institution process will be opaque.

Step 2'. The receiver then offers to the author the response B', namely that the request C' was judged by the institution's black box as not valid, without any evidence of the proceedings. Mind that at this step the receiver behaves as if it is a reporter of a past judgement, even if the institutional process happened after the submission of the argument by the author.

Step 1'. Finally, the receiver goes back one more step in time and produces A', a judgement decision: the thesis A is dismissed, with the excuse of a lack of resources of the institution (time, attention, interest), not the thesis. The reason for this lack is attributed to the author (maybe if the author would try harder, then the decision would be different, maybe if the author would use other venues for the discussion, etc).

As a consequence of this counter of the argument, the discussion ends, in such a way that the receiver does not have to provide reasonable further arguments to the author.

As an illustration, is reproduced here the answer to the submission of this article in the cs.AI, cs.HC sections of arXiv.org, a very reputable repository of articles which serves the research community by providing fast dissemination of research. Some times the streamlined submission process becomes an unsolicited editorial one, as they are not a journal, like in the case of the initial version of this article. (As an author who uses arXiv since a long time, this is the first time when I am part of this process, perhaps due to the unconventional nature of this article.) I wish to thank the institution for the provision of a real life counter attack of a valid argument by A summary.

The response is edited (only) to separate it into the 3', 2', 1' time reversed steps which were explained previously. Links and general statements were edited

out.

Step 3’. ”MOD-30319 [...] notification regarding [the initial version of this article]. Thank you for submitting your work to arXiv.””

Step 2’. ”We regret to inform you that arXiv’s moderators have determined that your submission will not be accepted and made public on [arXiv.org]. Our moderators determined that your submission does not contain sufficient original or substantive scholarly research and is not of interest to arXiv.”

Step 1’. ”arXiv moderators strive to balance fair assessment with decision speed. We understand that this decision may be disappointing, and we apologize that, due to the high volume of submissions arXiv receives, we cannot offer more detailed feedback. Some authors have found that asking their personal network of colleagues or submitting to a conventional journal for peer review are alternative avenues to obtain feedback.”

References

- [1] M. Buliga, Asemantic computing, in: chemlambda. (2022). chemlambda/molecular: Molecular computers which are based on graph rewriting systems like chemlambda, chemSKI or Interaction Combinators (v1.0.0). Zenodo. <https://doi.org/10.5281/zenodo.7306917> , [github source]
- [2] Y. Lafont, Interaction Combinators, *Information and Computation* **137**, 69–101, (1997)
- [3] M. Buliga, Artificial life properties of directed interaction combinators vs. chemlambda, [arXiv:2005.06060](https://arxiv.org/abs/2005.06060) (2020)
<https://mbuliga.github.io/quinegraphs/ic-vs-chem.html#icvschem>
- [4] M. Buliga, chemSKI with tokens: world building and economy in the SKI universe, [arXiv:2306.00938](https://arxiv.org/abs/2306.00938), (2023)
- [5] Phind, www.phind.com
- [6] Archive of Phind answers, <https://archive.ph/DUsn8>
- [7] M. Buliga, Argument from AI summary, (2023)
chorasimilarity.wordpress.com/2023/10/01/argument-from-ai-summary/