https://doi.org/10.5281/zenodo.2550304
UDC: 004.82.021:519.81

FORMAL AXIOMATIC SYSTEMS: A TRANSITION FROM M-MODE TO I-MODE

Festim Halili, Merita Kasa Halili, Valbon Ademi, Halim Halimi
University of Tetova – Department of Informatics, Tetovo Macedonia
E-mail(s): festim.halili@unite.edu.mk, merita.kasa@unite.edu.mk, valbon.ademi@unite.edu.mk, halim.halimi@unite.edu.mk

ABSTRACT

The Formal Axiomatic Systems are used in Artificial Intelligence and Mathematics to indicate any set of axioms, from which some or all axioms can be in conjunction to provide theorems. In this paper, we tend to combine and analyze FAS (Formal Axiomatic Systems) and Agents of Artificial Intelligence to enhance their intelligence, by promoting a transition from mechanical to intelligent mode. In addition, we will apply typographic and arithmetic methods to define isomorphism and its importance. It would be very easy to program a computer to generate theorem after theorem of a given system, however, if a machine could transit from the so called mechanical mode and use the intelligence, it would jump out of the formal axiomatic system and think autonomously. The recursive algorithm will be used to specify a model that would use the prior step to find the next one in a case study. The mapping between self referenced systems and formal axiomatic systems would help researchers differentiate the performances of artificial agents in their perceptions and actions in a certain environment.

Keywords: Artificial Intelligence, FAS, MIU System, Isomorphism, Recursive algorithm, agents.

Introduction

Self-reference in formal arithmetic can be compared to a formal method of self recognition in the mirror. The basis for this comparison is the role of the Gödel code in arithmetical self-reference [1]. Gödel first incompleteness theorem states that certain formal systems cannot be both consistent and complete at the same time. One could think this is easy to prove, by giving an example of a self-referential statement, for instance: "I am not provable". But the original proof is much more complicated: It was proved by composing a statement that indirectly referred to itself as 'This statement cannot be proved' - to be more precise, it says: 'The i-th proposition is not provable'.

ISBN: 978-608-66191-0-7
IMSC-2018, May 26, 2018. Tetovo. Macedonia.

By looking just at this sentence, it certainly isn't self-referential, but if we look at how all the propositions were numbered, we can see that is the number of the above proposition, so the self reference is not direct.

Is it what makes the theorem so important and the reason why the proof is so complicated - the fact that statements containing no direct self-reference whatsoever might still refer to themselves indirectly. Self-reference has a problem, if you want to think about it in terms of "I am not provable" sort of approach. A well-formed formula cannot refer to itself. Moreover, a formula cannot refer to the meta-theory (which is where proofs exist).

Emil Post in the 1920's (known as Post Production System) – defines the FORMAL AXIOMATIC SYSTEMS (FAS). In his work, Post observed that we can use normal systems as abstract version of formal axiomatic systems to represent decision procedures [2].

Douglas Hofstadter - introduces FAS by means of a very interesting puzzle the MU puzzle.

A Simple formal system, more exactly the MIU-system is presented, and the reader is urged to work out a puzzle to gain familiarity with formal systems in general. A number of basic notions are provided: strings, theorem, rule of inheritance, axiom, derivation and thinking or working within the system or outside the system [3].

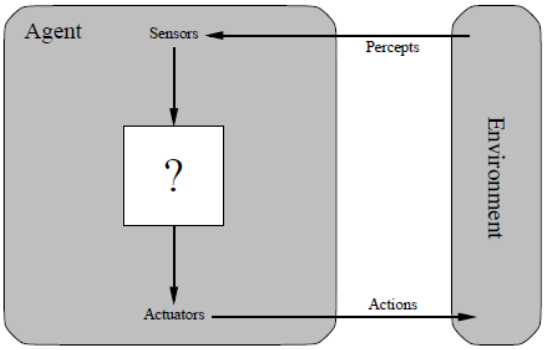
1. Artificial Agents – a pragmatic approach

Agents of Artificial Intelligence include humans, robots and softbots. They are acting in a specific environment and increasing the perception by using its sensors with intention to have a better outlook for the ambience where they act.

In general, the simplest agents have architecture with few elements included, which assist to percept the environment where they are acting. In the following figure, we can see that agent's percept the environment through their sensors and calculate that perception, afterwards using their actuators they make actions to improve the environment [4].

ISBN: 978-608-66191-0-7
IMSC-2018, May 26, 2018. Tetovo. Macedonia.

Figure no. 1 Architecture of an Agent



Source: Carlucci Aiello and Nardi, Artificial Intelligence Agents, lecture notes, 2009.

It is important to have a clear view of the structure of the agent, because we will have to take in consideration all these elements when building our speaking bot agent. It will be important to design the bot and also to design the environment in such a way that it can interact with it.

There are many tools to design a bot, but we prefer a software called Character builder, which is similar with the Adobe flash and it provides different opportunities to for creation of different agent faces and environments.

2. Designing Rational Agents

The agent function maps from percept histories to actions: f: P* → A, whilst the agent program runs on the physical architecture to produce the function.

Rationality is very important value in artificial intelligence field towards the agents. A rational agent chooses whichever action maximizes the expected value of the performance given the percept sequence to date.

Table 1. Non included elements in Rational Agent

Rational	≠	Omniscient
Rational	≠	clairvoyant
Rational	≠	successful

ISBN: 978-608-66191-0-7
IMSC-2018, May 26, 2018. Tetovo. Macedonia.

Table 2. Included elements in Rational Agent

Rational	=>	Exploration
Rational	=>	learning
Rational	=>	autonomy

To develop the rational agent we must specify the task environment, by implementing the PEAS method.

Performance measure: answer questions, think rationally, intelligent mode, comfort.
Environment: virtual environment
Actuators: no actutators
Sensors: keyboard, mouse, detect phrases.

It is important to mention that the environment influences the agents design. There can be several types of environments: Observable, Deterministic, Episodic, Static, Discrete and Single Agent.

In out previous paper work, we have constructed an Albanian speaking softbot, which was designed according to the PEAS methods and was constructed to think in intelligent mode. The applications used to establish such a softbot were: Microsoft Agent Control 2.0, AIML bot.dll file and C# programming language. In the continuing of this paper, we will differ between I-mode and M-mode, through a case study [4].

3. Recursive algorithms and the power of 2

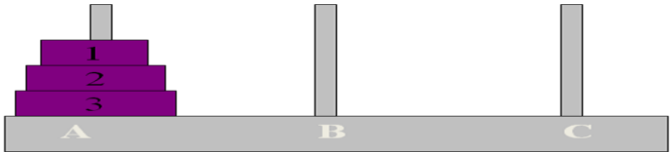
There have been developed many algorithms to solve different problems in informatics, but also for other areas of study. A recursive algorithm is an algorithm which calls itself with "smaller" input values, and which obtains the result for the current input by applying simple operations to the returned value for the smaller input.[1] We will use the recursive algorithms to solve the Hanoi towers problem and also define which mode of self reference system would be implemented. The Hanoi Tower, sometimes referred to as the mystery of the end of the world was first introduced by French mathematician Eduard Lucas. He was inspired by the Hindu legend regarding the history of a temple. The legends says that to the monks of the temple were given 64 golden disks, and their duty was to transfer all these disks from pattern A to pattern C (there were three patterns), the rule was that not a single larger disk should be above a smaller one while the transfer. The myth said that in case the

[1] Definition by the computer science lectures at the Old Dominon University.

ISBN: 978-608-66191-0-7
IMSC-2018, May 26, 2018. Tetovo. Macedonia.

monks finish this task, than the temple would vanish, but also the end of the world would come. The question is, how much time would the monks need to transfer 64 disks?

Figure no. 2. Three disks and three patterns



In this case we could use the Programming Logic (Prolog) to construct an algorithm that would solve this problem. The recursive model uses the previous step to find the next problem, meaning that if we want to know how many steps we would need to transfer 64 disks, in that case we would need to find the steps for 63 disks. In this case the recursive model would not help in finding the steps or the time to transfer the disks. However, recursion could help us generating more numbers with intention to find a suitable model to resolve this problem. Lets take S as the number of moves to transfer n-1 disks from pole A to C:

- For 1 disk, we need only 1 move to transfer it from A to C.
- For 2 disks, we need 3 moves: 2S+1=2(1)+1=3
- For 3 disks, we need 7 moves: 2S+1=2(3)+1=7
- For 4 dosls, we need 15 moves: 2S+1=2(7)+1=15 and so on

The power of number 2 is a fascinating fact in finding the correct and easiest model to solve this issue.

Table 2: 2^n – the power of number 2

Nr of disks	**Nr of steps to take**
1	$2^1 - 1 = 2 - 1 = 1$
2	$2^2 - 1 = 4 - 1 = 3$
3	$2^3 - 1 = 8 - 1 = 7$
4	$2^4 - 1 = 16 - 1 = 15$
5	$2^5 - 1 = 32 - 1 = 31$
6	$2^6 - 1 = 64 - 1 = 63$
7	$2^7 - 1 = 128 - 1 = 127$

ISBN: 978-608-66191-0-7
IMSC-2018, May 26, 2018. Tetovo. Macedonia.

In this case, the formula to find the steps to transfer disks from pole A to pole C would be: 2^n-1. What would be the number of steps to transfer 64 disks? According to the formula it would be 2^{64}-1, and we would have a very large number: 18,446,744,073,709,551,615 steps. If the monks work day and night and they perform one move in a second, we would need 580 billion years to finish the task. Its much more than many scientific predictions regarding the life of the lasting of the sun system [5].

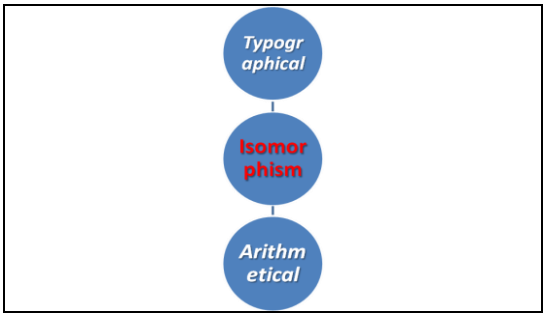
The implementation of the algorithm in Prolog is:

```
Solution(S) :- bejSolution(S, 'A', 'B', 'C').
        bejSolution(0, _ , _ , _ ) :- !.
        bejSolution(S, A, B, C) :-
                S_1 is N-1, bejSolution(S_1, A, C, B),
                move(A, C),
                bejSolution(N_1, B, A, C).
        move(From, To) :-
                 write([move, From, -->, To]), nl.
```

4. MIU System - M-mode and I-mode

In the MIU system, the isomorphism takes the typographical system in to an arithmetical one (actually invertible, hence it is always possible to come back to the original FAS). Gödel numbering is one example of an useful isomorphism to study FAS by transforming it into TNT.

Figure no. 3 Role of isomorphism



Here we will have to introduce the formal system in the form of a little puzzle. "Can you produce MU?" is the puzzle. To begin with, we will be supplied with a string

ISBN: 978-608-66191-0-7
IMSC-2018, May 26, 2018. Tetovo. Macedonia.

(which means a string of characters), and that string will be MI. A string means a string of characters. The strings of the MIU-system consists of the letters M, I and U in any combination and number. Below are some examples of strings:

MU UIM
MIU MIIUIIMU
MUUMMU IIIIIIUM

It is important to mention here that, although these are strings, these are not in our possession to use them during our manipulations with the rules that will be given. These are string that are created by the rules of this formal axiomatic system.

Rule 1: If you possess a string whose last letter is I, you can add on a U at the end. Example: MII – MIIU
Rule 2: Suppose you have Mx. Then you may add Mxx to your collection. Here the letter 'x' stands for any string ('x' itself is not a part of the MIU-system). Example: MIU – MIUIU
Rule 3: If III occurs in one of the strings in your collection, you may make a new string with U in the place of III.
Example: MIIII – MUI
Rule 4: If UU occurs inside one of your strings, you can drop it. Example: MUUI – MI

In fact it would be very easy - to program a computer to generate theorem after theorem of the MIU-system. The computer-program could do this in the following methodical way, - an algorithm for creating theorems of the MIU-system: tree of theorems.
Step 1: Apply every applicable rule to the axiom MI. This yields two new theorems: MIU, MII.
Step 2: Apply every applicable rule to the theorems produces in step 1.
Step 3: Apply every applicable rule to the theorems produces in step 2.

If we stop acting like a machine and if we start using our intelligence we can notice:

- that all theorems would begin with M

ISBN: 978-608-66191-0-7
IMSC-2018, May 26, 2018. Tetovo. Macedonia.

- the property that they make each new theorem inherit its first letter from an earlier theorem; ultimately, then, all theorems' first letter can be traced back to the first letter of the sole axiom MI - and that is a proof that theorems of the MIU-system must all begin with M.

What we did here was using our intelligence to jump out of the formal system. It is very important when studying formal systems to distinguish between working *within* the system, behaving like a machine (M-mode), and making statements or observations *about* the system, using your intelligence (I-mode).

Conclusion

One of the most significant features of artificial agents and artifacts with good governance is their commitment to maintain a high degree of intelligence that is demonstrated through continuous feedback and thinking about the system is the I-mode. Engineers are able to create algorithms to solve different mathematical or other problems in M-mode, however we still lack in creating agents in fully I-mode.
However, many scientist believe that if we reach to the point where the self-reference systems would implement the I-mode to other systems, and agents would be able to create other agents, or with other words to reproduce themselves, then humanity could be at risk of not having full control to these kind of systems.
We should use our intelligence to ease the life of human societies, by creating intelligent agents and robots, but also taking care of the possible casualties and bugs that we might create unintentionally or even intentionally by irresponsible people.

References

1. Panu Raatikainen, Gödel's Incompleteness Theorems, in the Stanford Encyclopedia of Philosophy, November 11, 2013.
2. Dov M. Gabbay and John Woods, Handbook of the History of Logic. Volume 5. Logic from Russell to Church, 2008 Elsevier BV.
3. Theo Johnson-Freyd, Math 105-6: Freshman Seminar: Theories Of Mind And Mathematics, Northwestern University, 2014.
4. Festim Halili, et.al, Combining AIML, C# and Agent Control 2.0 to produce an Albanian Speaking bot, in International Conference Proceedings ISTI, 2013, Tirana Albania.
5. Uri Levy, The Magnetic Tower of Hanoi, Atlantium Technologies, Har-Tuv Industrial Park, Israel 2010

ISBN: 978-608-66191-0-7
IMSC-2018, May 26, 2018. Tetovo. Macedonia.