



# VISUALIZATION OF FORMAL SEMANTICS - POSSIBILITIES OF ATTRACTING FORMAL METHODS IN TEACHING

William Steingartner<sup>1</sup>,  
Matúš Jankura<sup>1</sup>,  
Davorka Radaković<sup>2\*</sup>

<sup>1</sup>Technical University of Košice,  
Faculty of Electrical Engineering and  
Informatics,  
Košice, Slovakia

<sup>2</sup>University of Novi Sad,  
Faculty of Sciences,  
Novi Sad, Serbia

## Abstract:

The goal of formal semantics of languages is to reveal the nature of a language beneath its syntactic surface. One of the popular and frequently used methods is operational semantics. Its categorical representation using coalgebras is a new approach to semantics. Current trends show that software visualization of various calculations contributes to clarity and easier understanding. To support students' interest in semantic methods, we designed and prepared a software tool whose task is to visualize computations in coalgebraic operational semantics.

## Keywords:

Coalgebraic Operational Semantics, Formal Semantics, Software Visualization, University Didactics, Education.

## INTRODUCTION

The global COVID-19 pandemic has directly influenced the change of the basic worldwide concept of head-to-head teaching in schools and universities [1,2,3,4]. It was necessary to react quickly and reorganize teaching to be all online, using online platforms as virtual classrooms. Initially, all forms of online communication were used, from Skype, Viber, through Zoom, Discord, Webex platforms to Google classrooms, only to establish communication with students. Now, after more than a year of constant online teaching, the choice of platforms has stabilized and schools have mostly decided to use one platform in their teaching. In this way, the education was implemented in the home environment.

Besides that, the COVID-19 pandemic has provided an opportunity to strengthen and to spread digital learning, in last decades with a development of ICT, contemporary education has already started with use of visual applications which facilitate comprehension of the teaching subjects [5,6,7]. Dynamic Geometry Software (DGS) is widely used for visualisation in teaching and learning mathematical topics [8, 9, 10, 11, 12].

## Correspondence:

Davorka Radaković

## e-mail:

davorkar@dmi.uns.ac.rs



The KEGA project (cited in Acknowledgment) is focused on improving and making teaching methods more attractive applied in the lecture and laboratory part of the subject. Its topicality is growing especially in the present, when the current full-time teaching replaced by a combined and distance form. Our goal is to apply modern software solutions in this course as a significant help in lecture and laboratory activities during the teaching of the subject, in all forms teaching (full-time, part-time, distance) and to a large extent also support independent preparation of students.

In the future, we want to apply the proposed software solution for the evaluation part of the course (tests, exams). During the work on the project, we focus on the preparation and implementation of an integrated package of programs that will allow illustrative and understandable use of semantic methods. In this way we will achieve a greater clarity of the applied procedures and principles in the teaching of the subject.

The paper is organized as follows: The second section provides an overview of related work. Section 3 introduces our coalgebraic operational semantics translator which translates Jane language into coalgebraic operational semantics. In the Section 4 we give the architecture of most important parts of application. Last section concludes the paper.

## 2. RELATED WORK

The practical usage of the coalgebras in computer science used for teaching of formal methods for young software engineers is given in [13]. Operational semantics was introduced by Plotkin in [14]. Structural operational semantics of imperative language Jane is presented in [15].

Due the pandemic, e-learning become the main form of educational activity at all levels of education [1,2,3,4]. On the other side, when e-learning was not mandatory, as it was in during last year, it was often associated with problems resulting from the lack of interactivity and the simple transmission of knowledge from the computer screen. To overcome this shortage, we have developed web application where students can see how semantics works: in coalgebraic (operational) semantics, the Q endofunctor performs one calculation step for each application [16].

## 3. COALGEBRAIC OPERATIONAL SEMANTICS TRANSLATOR

Coalgebraic operational semantics translator is a web application, which translates extended version of Jane language into text and diagram representation of coalgebraic operational semantics. The translation is done in the following steps:

1. Code input
2. Lexical and syntax analysis
3. Code structure
4. Showing result

First, we have to write text input (see Listing 1), that should be a code in Jane language.

```
var x;  
  
input x;  
x:=x+1;  
print x
```

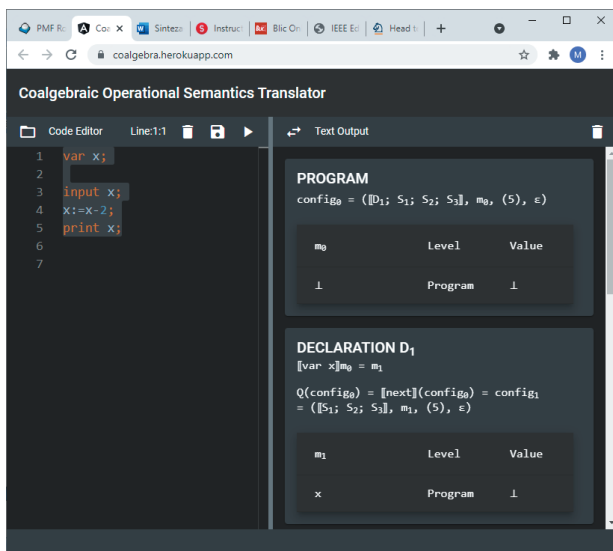
Listing 1 - An example of a code written in Jane.

The input goes through lexical and then syntax analysis. Basically, if input is not written in Jane language or has some lexical or syntax errors, we cannot translate the code into coalgebraic operational semantics. If there are no errors in user typed code input, then we can do semantics analysis and go further with translation.

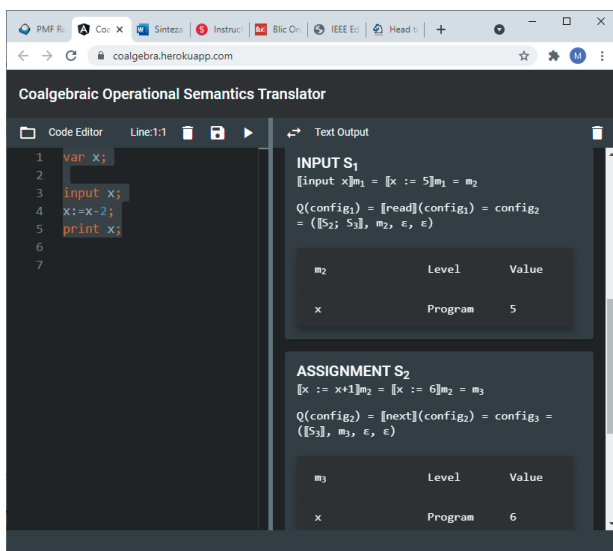
With syntactically correct code we create a code structure, which is our representation of code output. We analyse each line of code and create their instances. Each line of code instance has all information about that line of code. For example, from assignment statement we keep information of variable, in which we assigned an arithmetic expression and we also keep information about the arithmetic expression as well, so in next step we have all needed information for translation. We create code structure even if code had grammar errors, but of course we won't have information about execution of lines of code, we will instead have information about those grammar errors, what caused them and on which line and column of code the given error occurred.



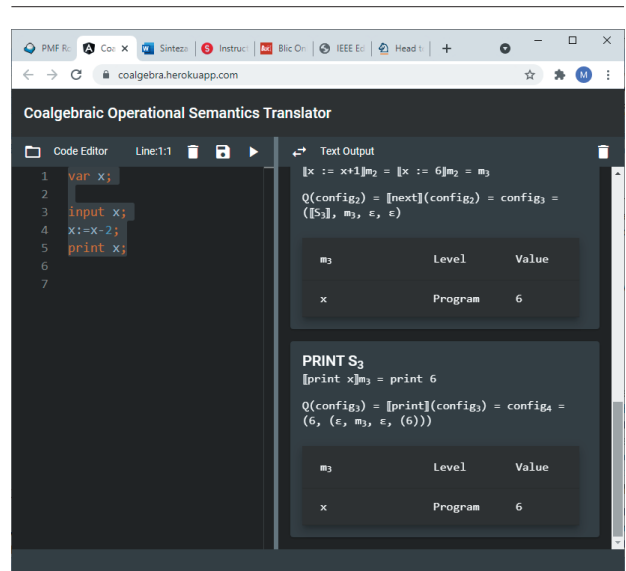
In the case of getting grammar errors, we show them to user, so the user can fix his code and do the process all over again. Otherwise, we translate code structure into coalgebraic operational semantics and show its text, as shown in Figure 1, and diagram representation as shown in Figure 2.



a) Program and Declaration



b) Input and Assignment



c) Print

Figure 1 – Text output for x=5.

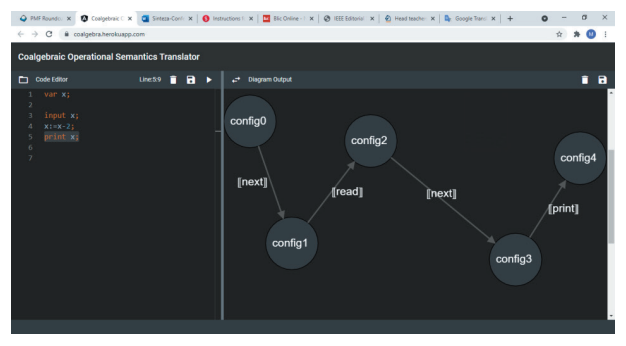


Figure 2 – Diagram output for x=5.

## 4. ARCHITECTURE

Behind steps explained in previous section is the architecture separated in two logics: front-end and back-end (Figure 3). On front-end we have implemented code editor powered by Monaco editor [17] created by Microsoft and derived from their Visual Studio Code editor. As the code is inserted, it is sent to back-end, and the code structure is obtained. When the result gets back, it is shown to user. If there are errors, the interactive error message containing information of issue that caused the error is shown together with a line and a column, where the error occurred. When there is no error, it is shown the translation into coalgebraic operational semantics.



The text representation is also interactive; all executed lines of code are clickable, highlighting the lines of code, which were translated. In diagram representation of coalgebraic operational semantics all the configurations and steps between them are shown, with clickable nodes, also highlighting the lines of code, which were translated.

Back-end part consists of analysing the code input with lexical and syntax analysis done by ANTLR created by Terrence Parr [18]. ANTLR also gives us the opportunity to easily attach semantics analysis to the process. We have created grammar for ANTLR given in [19].

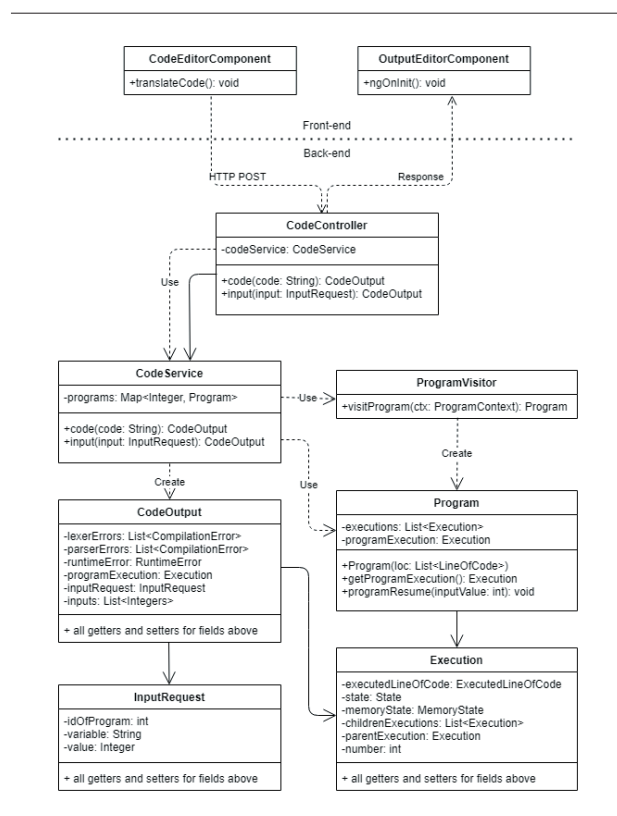


Figure 3 – Architecture.

## 5. CONCLUSION

In this article, we presented the results achieved in the field of research and development of software tools, which contribute to the attractiveness of teaching semantic methods and which contribute to the clarity of the semantics and its deeper understanding. Educating young IT experts is always a challenge: how to keep up with trends in information technology and how to pass on information to students. Formal methods have their irreplaceable place in software engineering. Almost all formal methods have their basis in the formal semantics of languages.

Thanks to this, the education of future software experts is always supported by a course on formal semantics. In the field of formal semantics, there are several directions, some of which require deeper mathematical knowledge, other methods are also suitable for developers who work with a mathematical basis only minimally.

The use of category theory highly contributes to a solid mathematical basis for formal methods. Categories have a strong expressive ability and allow to model relationships between mathematical structures in a very elegant way. The software tool we presented in this work allows us to visualize and model the coalgebraic operational semantics defined in [20].

We want to further apply the results from the modeling of categorical semantics in teaching. Experience shows that the software has a very good response from students. Students understand the principles of methods very well, if they do not work exclusively with mathematical relations, but can visualize calculations in semantics on an interactive basis.

We are of the opinion that the development of (own) visualization software tools will contribute to the attractiveness of teaching formal methods for students and engineers in practice, especially during distance learning, but also during full-time teaching, as interactive visualization brings some dynamism to the teaching process.

## 6. ACKNOWLEDGEMENTS

*This work was supported by project KEGA 011TUKE-4/2020: "A development of the new semantic technologies in educating of young IT experts".*

## REFERENCES

- [1] K. Potyrała, N. Demeshkant, K. Czerwiec, B. Janczarz-Łanczkowska and Ł. Tomczyk, "Head teachers' opinions on the future of school education conditioned by emergency remote teaching," *Educ. Inf. Technol.*, 2021.
- [2] M. Bakator and D. Radosav, "Managing Education in the COVID-19 era," in *International Conference on Information Technology and Development of Education – ITRO 2020*, 2020.
- [3] E. M. Mulenga and J. M. Marbán, "Is COVID-19 the Gateway for Digital Learning in Mathematics Education?," *Contemporary Educational Technology*, p. e269, 2020.



- [4] S. Pokhrel and R. Chhetri, "A Literature Review on Impact of COVID-19 Pandemic on Teaching and Learning," *Higher Education for the Future*, pp. 133-141, 2021.
- [5] T. Dragon and M. Klement, "Use of Web and Mobile Applications as an E-Learning Tool in an International Context," in *ICERI2020 Proceedings*, 2020.
- [6] S. Jablonski, A. Caldeira and A. Moura, "WTM Verlag für wissenschaftliche Texte und Medien," pp. 71-78, 2020.
- [7] I. Gurjanow, J. Zender and M. Ludwig, "MathCity-Map – Popularizing Mathematics around the Globe with Math Trails and Smartphone," in *In M. Ludwig, S. Jablonski, A. Caldeira, & A. Moura (Eds.), Research on Outdoor STEM Education in the digital Age*, 2020.
- [8] M. Hohenwarter, Z. Kovács and G. Ancsin, "GeoGebra goes mobile," vol. 5, no. 2, p. 160–168, 2011.
- [9] Đ. Herceg, D. Radaković, M. Ivanović and D. Herceg, "Possible Improvements of Modern Dynamic Geometry Software," *Computer tools in education*, vol. 2, p. 72–86, 2019.
- [10] D. Herceg, D. Radaković, D. Herceg and V. Herceg Mandić, "Subject-specific components in dynamic geometry software," *International Journal for Technology in Mathematics Education*, vol. 26, no. 2, pp. 97-102, 2019.
- [11] M. Hohenwarter, J. Janičić, Z. Kovács, I. Petrović, T. Recio, S. Weitzhofer and F. Botana, "Automated theorem proving in *GeoGebra: current achievements*," *J. Autom. Reason*, vol. 55, pp. 39-59, 2015.
- [12] L. Diković, "Applications GeoGebra into teaching some topics of mathematics at the college level," *Comput. Sci. Inf. Syst.*, vol. 6, pp. 191-203, 2009.
- [13] W. Steingartner, D. Radaković, F. Valkošák and P. Macko, "Some properties of coalgebras and their rôle in computer science," *Journal of Applied Mathematics and Computational Mechanics*, vol. 16, no. 3, pp. 145-156, 2016.
- [14] G. Plotkin, "An Structural Approach to Operational Semantics," Technical Report DAIMI FN-19, Computer Science Department, Aarhus University. , 1981.
- [15] W. Steingartner, M. A. M. Eldojali, D. Radaković and J. Dostál, "Software Support for Course in Semantics of Programming Languages," in *Proceedings of the 14th International Scientific Conference on Informatics, INFORMATICS 2017*, 2017.
- [16] [Online]. Available: <https://coalgebra.herokuapp.com/>. [Accessed 10 06 2021].
- [17] [Online]. Available: <https://microsoft.github.io/monaco-editor/>. [Accessed 10 06 2021].
- [18] T. Parr. [Online]. Available: <https://www.antlr.org/>. [Accessed 10 June 2021].
- [19] M. Jankura, "The visualization of a coalgebraic semantics of imperative languages," Technical University of Košice, Košice, 2021.
- [20] W. Steingartner, V. Novitzka and W. Schreiner, "Coalgebraic Operational Semantics for an Imperative Language," *Computing and informatics*, vol. 38, no. 5, 2019.