

Visualization module solving logical problems for the FORMULA system

Ksenia Panova

1. INTRODUCTION

We have designed the module of visualization of solutions obtained via the smt solver Formula. Solvers for satisfiability modulo theories (SMT) check the satisfiability of first-order formulas containing operations from various theories such as the booleans, bit-vectors, arithmetic. A thorough consideration of various methods of visualization was made. As a result of studies the declarative programming language for solutions of logic problems such as Einstein task or the problem of queens was developed.

2. PROBLEMS

The developed system is designed for data visualization. The purpose of the study is to consider an information visualization solutions of logical problems derived from the FORMULA system, which is fundamentally different from existing visualization tools. The most obvious advantage of the new system is that the developed module operates with a much smaller amount of information, which saves time and expense of resources for the visualization process.

The main task of the investigation is to design an integrated module of visualization solutions and the results of logical tasks.

The developed module should maintain the following functions:

1. Integration module logical solution FORMULA tasks, analysis of the results of using it
2. Display of static solutions with support for the following types of visualization:
 - (a) Table
 - (b) Counts
 - (c) Vectors
3. Display of the process of solving problems by means of a dynamic visualization:

- (a) Moving objects
- (b) Time Management

3. ARCHITECTURE DESIGN

The project structure is shown in Fig. 1 The system is a program running in Windows. It analyzes configured via FORMULA file, and loads the content to the respective variable depending on the type of task and the format:

- For a static display in a table - table
- For dynamic display - sequence

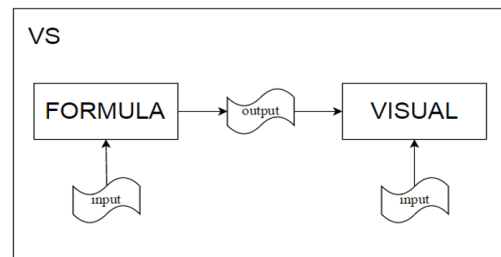


Figure 1. Project structure

4. LANGUAGE MODULE

Starting the system designed by the programming language is accompanied by its conversion into a form that is meaningful to the computer. Therefore, we need a program that reads the text written in one language (the source) and translates it into an equivalent text in another language.

In order to provide this, compiler that converts a program written in the language of visualization, in a program written in C++ was developed.

For that, lexer performs lexical analysis which is the process of converting a sequence of characters into a sequence of tokens (strings with an identified "meaning"). Lexical analysis was performed using parser generator

"Lex".

After lexical analysis is the syntactic analysis. Parsing or syntactic analysis is the process of analysing a string of symbols, either in natural language or in computer languages, conforming to the rules of a formal grammar. The purpose of syntactic analysis is to determine the structure of the input text. This structure consists of a hierarchy of phrases, the smallest of which are the basic symbols and the largest of which is the sentence. The structure can be described by a tree with one node for each phrase. Basic symbols are represented by values stored at the nodes. The root of the tree represents the sentence. an example is shown in Fig. 2. Syntactic analysis was performed using generator "Yacc", which provides automatic construction of grammatical analysis procedures.

As a result of the joint compilation of the generated lexer and parser , executable file visual.exe is created for handling of code with new syntax.

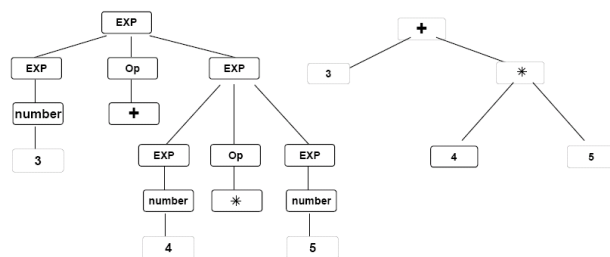


Figure 2. Example of syntactic analysis

5. CONCLUSIONS

As a result of studies the integrated module of visualization solutions and the results of logical tasks was designed. Also visualization language was developed. For that rules and a new programming language syntax have been formulated, on the basis of which carried out the lexical, syntactic and semantic analysis. Graphical system module is also designed to visualize input data in a predetermined format, namely in the form of static and dynamic table moving objects.

References

- [1] M. Kaufmann, Information Visualization: Perception for Design, 2004.
- [2] A. Aho, M. Lam, R. Sethi, LCompilers: Principles, Techniques, and Tools, 2008.
- [3] M. Khan, S.S. Khan, Data and Information Visualization Methods, and Interactive Mechanisms: A Survey, 2010.
- [4] D. Kulikov, Means, solutions and approaches to data visualization in medical information systems, 2009