**Francisco J. Almeida-Martínez**
**Jaime Urquiza-Fuentes**
**Antonio Pérez-Carrasco**

# Building LL (1) parsers with VAST. Second usability and educational evaluation.

## Number 2011-06

**Serie de Informes Técnicos DLSI1-URJC**
**ISSN 1988-8074**
**Departamento de Lenguajes y Sistemas Informáticos I**
**Universidad Rey Juan Carlos**

# Index

1. Introduction .....3
2. Description of the evaluation .....4
2.1. Subjects.....4
2.2. Experimental design.....5
2.3. Tasks.....5
2.4. Protocol.....6
3. Results….....7
3.1. Instructors observations.....7
3.2. Answers to questionnaires.....8
3.3. Educational effectiveness.....10
3.4. Solutions to the practices.....11
4. Conclusions.....12
References.....13
Appendix.....14

# Building LL(1) parsers with VAST. Second usability and educational evaluation

Francisco J. Almeida-Martínez, Jaime Urquiza-Fuentes and Antonio Pérez-Carrasco

LITE – Laboratory of Information Technologies in Education
Technical Superior School of Computer Science Enginnering
Rey Juan Carlos University
{francisco.almeida, jaime.urquiza, antonio.perez.carrasco}@urjc.es

**Abstract.** VAST is an educational tool to use in subjects as compilers and language processors which allows to display the syntax tree. Its main feature is that can be used independent from the generation tool used to build the parser. This makes possible to visualize and animate the syntax tree generated by the parser. In this work we describe the second usability and educational evaluation using VAST comparing the results with ANTLR and ANTLRworks.

## 1 Introduction

VAST [1] is an educational tool to be used in subjects as compilers and/or language processors. The main advantages of VAST are: its design in order to be independent from the generation tools, it allows the student to watch the syntax tree (including the syntax error recovery) and it has an user interface –VASTview- to manage huge syntax trees easily.

In this evaluation we have used ANTLRworks [2], a tool which allows to display parsers LL (1) previously built with the parser generator ANTLR [3].

This technical report describes the usability-educational evaluation and the instructor's observations while the students used the tool. The evaluation has been performed in the subject Language Processor in the course 2009-2010 in the Computer Science degree at the Rey Juan Carlos University.

In Table 1 it is shown a list of the features available in VAST. The first column refers to the features available in the current evaluation. The second and the third column show the features analyzed in each evaluation.

| Aspect | 2008-2009 | 2009-2010 |
|---|---|---|
| Vis. Syntax tree | ✕ | ✕ |
| Subtree resume | ✕ | ✕ |
| Zoom | ✕ | ✕ |
| Visual configuration | ✕ | ✕ |
| Inicial configuration | | ✕ |
| Edit input stream | | ✕ |
| Stack view | ✕ | ✕ |
| Grammar view | | ✕ |
| Global view | ✕ | ✕ |
| Text explanations | | ✕ |
| Manual reproduction | ✕ | ✕ |
| Automatic reproduction | ✕ | ✕ |
| Parser import | | ✕ |
| Manual annotation | ✕ | |
| Manual compilation | ✕ | |
| Syntax error visualization | | |
| Multiplatform | | ✕ |

**Table 1. Features available in VAST**

The rest of the report is structured as follow. In section 2 it is described the evaluation, the subjects, the method and the protocol used. In section 3 we describe the results, taking into account the instructors' observations, the answers to the questionnaire, the pre-postest and the solutions to the practices. Finally, in section 4 it is shown the conclusions of this work.

## 2 Description of the evaluation

In this section it is described the evaluation. We describe the students which participated in the evaluation, the experimental design, the protocol and the task performed during the experiment.

### 2.1 Subjects

In this evaluation participated 40 students of the subject *Language Processor* at the Rey Juan Carlos University. The participation was based-incentive in 2% over the final mark if they passed the subject and were in all the sessions. The gender distribution was not balanced, so the 15% (6/40) of the students were women.

### 2.2 Experimental design

This evaluation was designed as an observational and usability study including the educational effectiveness. In order to create different groups the students were divided using a pretest of knowledge (see appendix A). Besides, we asked the students for their opinion of different aspects of VAST which included: ease of use, learning support, quality of the tool, satisfaction and other aspects (see appendix C, D and E).

As we have said before, students were divided in different groups. We used the marks of the pretest to divide the students, but the assignment to the corresponding group was random. The first group has to use only the generation tool ANTLR without using any visualization tool. The second group used the tool ANTLRworks. Finally, the third group used VAST.

The independent variable was the tool used: VAST for the treatment group, ANTLR for the control group 1 and ANTLRworks for the control group 2. The dependent variables were focus in the students' opinion about four fundamental aspects: ease of use, learning help, quality of the tool and personal satisfaction.

### 2.3 Tasks

The tasks performed by all groups have to be documented using visualizations and text explanations. In the control group 1 the visualizations have to be created by students; in the control group 2 and treatment they have to use the visualization built by ANTLRworks and VAST respectively. The evaluation consisted in one exercise about the way of working of LL (1) parsers. Before the session we explained in each group how to generate visualizations. In case of the control group 1 (ANTLR) one alternative was to display the syntax rule used. We gave the specification of the grammar to the students following the ANTLR notation.

**Control group 1 (ANTLR)**. Students have to run the generador from the command line. If the process was correct, they had to compile manually the parser, which required to configure the Java CLASSPATH correctly. Next they have to check if the solution was correct without using visualizations.

**Control group 2 (ANTLRworks)**. Students have to load the given specification in the ANTLRworks environment. In order to obtain the static visualizations generated by their parsers, they had to use the interpreter. However, to watch the animation of the syntax process they had to use the debugger. In both cases, it was necessary to write an input stream, the starting rule and the symbol of the final line according to the platform.

**Treatment group (VAST).** In this group students had to import the specification into VAST. Once the importation process finished, they had to write an input stream to run the parser from the VAST's user interface.

In the exercise, students had to change the precedence of the binary operators in the given grammar. The original specification was as follow:

S ::= FN
N ::= +FN | - FN | * FN | / FN | λ
F ::= id | cte | (S)

The new grammar had to give more precedence to the parenthesis. Next, one lower precedence to the operators * and / but both the same. In case of the operator – it has a lower precedence that the previous ones. Finally, the lowest precedence was to the operator +. In all cases the associability was on the right.

The solution to this exercise was a grammar including: one or some screenshots where the precedences were shown, the input stream/s used to generate the visualization/s and the text explanation.

### 2.4 Protocol

In this section we describe the protocol used in the evaluation and the order of the different tasks performed. Two weeks before the experiment, we did the pretest (see appendix A). From the results we built two control groups and one treatment group. One week before the session we gave a laboratory session in order to make the students familiar with the generator ANTLR. The evaluation session lasted two hours. Firstly the instructors gave a tutorial of the corresponding visualization tool (treatment and control group 2). From this point students had to work in the exercise of the evaluation. At the end of the session we asked the students for answering an opinion questionnaire of the tool used. Finally, one week after the evaluation, we performed the postest of knowledge. In Table 2 we show a resume of this protocol.

| Treatment group | Control group 1 | Control group 2 |
|---|---|---|
| Pretest of knowledge | | |
| ANTLR session | | |
| VAST tutorial | | ANTLRworks tutorial |
| Exercise | | |
| Questionnaire | | |
| Postest of knowledge | | |

**Table 2. Protocol of this evaluation**

## 3 Results of the evaluation

In this section we describe the resutls obtained during the evaluation. During the session the instructors observed how the students worked with the tools. The results are divided into instructor's observations, answers to the questionnaire, educational effectiveness and solution to the practices.

### 3.1 Instructors' observations

**Control group 1 (ANTLR)**. In this group the instructors saw different behaviours of the students. On the one hand it was interesting to see if any students used visual representation to watch the syntax tree. In this case one student used *Ms Paint* and another paper to draw the syntax tree. On the other hand, there were two students who had problems with Java. The rest of the students who had used previously Java, 6 had problems to configure correctly the CLASSPATH. According to the subject, 11 students looked for information (class theory, internet). Besides, 3 students had difficulties to understand correctly the precedence of operators. In relation with the ANTLR parser generator, students had different problems, so 8 students had problems with recursive rules in axiom level. Finally, 2 students had problems to define correctly the end of file in the grammar.

**Control group 2 (ANTLRworks)**. Using this tool the instructors observed 4 fundamental aspects. 6 students went to the evaluation after the rest of the class. 6 students used the debugger to check if their solution satisfied the exercise precedence. Besides, 11 students used paper to build the visual representation of the syntax tree according to the grammar proposed in their solution.

**Treatment group (VAST).** Students used different functionalities of the tool although it was not the objective of the evaluation. We detected that in the importation process one student obtained compilations error due to a bad initial configuration. We observed that 4 students used laptops with different operating system (2 Ms Windows, 1 Mac OS and 1 Ubuntu). There were 4 students that had problems to understand correctly the precedence of operations, mainly they did not understand when an operator had lower or higher precedence. There were 2 students that used paper to represent the syntax tree.

### 3.2 Answers to questionnaries

The answers to the questionnaire are divided in the marks obtained in different aspects of the tool used and in the comments (positive and/or negative) given by the students.

All the questions had to be answered using a *Likert* scale with 5 points being 1 the lowest mark, 3 without opinion or middle mark and 5 the highest mark. In Table 3 we show the average results obtained for the dependent variables with an analysis of significant differences.

In **question** 1 we asked the students to identify which parts of the tool were difficult to use. In the control group 1 (ANTLR) they identified different aspects: the necessity of using an external editor to implement the grammars, the configuration of the environment variables (necessary to use Java) and the use of the command line to generate and compile the parser. In control group 2 (ANTLRworks) there were some difficulties with the debugger (3 students), the parser's stack and the need of defining the initial symbol. In case of the treatment group (VAST) students pointed out:

problems with long names in non terminals symbols, difficulties to perform the initial configuration, the fact of hiding some functionalities and the import process which was considered as very boring

In **question 3** we asked the students to identify which parts had more or less quality in each tool. The control group 1 identified with high quality aspects the processing of the input stream and the edition of the grammar using any editor (3 students). In case of low quality they pointed out the use of the command line (2 students), the configuration of environment variables to compile and the time required to perform all changes and recompile. In control group 2 students found as aspects of high quality the visualization of the syntax tree in the static way (4 students), the grammar editor (2 students), the debugger (4 students), the interpreter (3 students) and the syntax tree's building process (2 students). As aspects of low quality they mentioned the visualization of the parser's stack (4 students), the way of specifying the grammar, the debugger due to the fact of having a port free (3 students), difficulties to display big syntax tree, the edition of the input stream and the error messages given by the tool. According to the treatment group as aspects of high quality they mentioned the building process of the syntax tree (3 students), the possibility of editing the input stream from the user's interface, the parser's stack and the synchronization of all views. In this case as aspects of low quality they identified the user's interface, the importation and the lack of error messages to the user.

In **question 6** we asked the students to identify which functionalities they would include in the tools. In the control group 1 they mentioned the need of showing the rule used (4 students), a graphic user interface and IDE (2 students) and the visualization of both the syntax tree and parser's stack (12 students). In the control group 2 students identified the improvement of the debugger changing the reproduction step by step (2 students), include the line number in the grammar specification, allow to export the visualizations of the syntax tree, improve the aspect of the syntax tree and remark the syntax rule used. In the treatment group they pointed out the necessity of saving the last importations performed. Besides they identified the necessity of improving the error messages given, the grammar editor (3 students), include the functionality of exporting animations of the syntax tree (2 students) and the help menu.

In **question 7** we asked about the functionalities which were not necessary in the tools. In the control group 1 they identified that the compilation was performed in two steps (lexical and syntax). Note that this was performed in this way in order to keep a relation with next practices. In the control group 2 they pointed out the graphic representation of the productions (3 students) and the debugger. In the treatment group they identified the global view and the functionality of resume/expand subtrees.

| Student's opinion | Average of control group 1 | Average of control group 2 | Average of treatment group |
|---|---|---|---|
| **Easy of use** | | | |
| General | 3.39 | 4.63 | 4.06 |
| Parts average | 3.28 | 3.89 | 4.05 |
| **Learning support** | | | |
| Building process | | | |
| Input stream | 3.22 | 4.17 | 4.24 |
| Stack | 3.56 | 3.71 | 3.53 |
| | 2.94 | 2.88 | 3.29 |
| **Quality of the tool** | | | |
| General | | | |
| Parts average | 2.94 | 3.92 | 3.82 |
| | 3.20 | 3.71 | 3.83 |
| General satisfaction | 2.94 | 4.25 | 3.76 |

**Table 3. Answers to questionnaires**

In **question 8** we asked the students to comment the positive aspects of the tool, especially if they had not mentioned yet. In the control group 1 they identified the similarity between the lexical and syntax specifications, the ease of installation, the speed of execution and the possibility of making scripts to make the process automatic. In the control group 2 they identify the functionality of pointing out the text in the editor, the execution step by step (2 students), the possibility of verifying if a grammar satisfied the LL (1) conditions, "it is better that using paper and pen..." and the visualization of the syntax tree. In the treatment group they mentioned that it was not necessary use ANTLR, ease to understand how the syntax tree was built and the visualization step by step.

Finally, in **question 9** we asked the students to comment the negative aspects of the tools. In the control group 1 they mentioned the necessity of learning to use the tool. Besides, when an error was detected, the causes were not clear (2 students). In the control group 2 they pointed out the error messages given by the tool (2 students) and the use of a static port to use the debugger (2 students). In the treatment group they mentioned the lack of tooltips.

### 3.3 Educational effectiveness

The educational effectiveness is focused on the analysis of the results from the differences between postest and pretest. The analysis of the data was performed fragmenting the students in different groups to make easier the study. We created a total of 5 groups (A-E). The students in groups A and B were considered very good or

good; C normal; D and E bad or very bad. Using these groups we obtained the results of Table 4, Table 5, Table 6 and Table 7.

| Level | G. Control 1 | G. Control 2 | G. Treatment | Stats |
|---|---|---|---|---|
| Knowledge (K) | 0.11 | 0.13 | 0.17 | An(2)=0.08, p=0.93 |
| Comprenhension (U) | 0.10 | -0.05 | -0.13 | X2(2)=1.98, p=0.37 |
| Aplication (Ap) | 0.02 | 0.03 | 0.21 | X2(2)=3.37, p=0.19 |
| Syntethis (S) | 0.19 | 0.14 | 0.17 | An(2)=0.08, p=0.93 |
| **Total** | 0.11 | 0.06 | 0.12 | An(2)=0.32, p=0.73 |

**Table 4. Results of educational effectiveness for groups A and B**

| Level | G. Control 1 | G. Control 2 | G. Treatment | Stats |
|---|---|---|---|---|
| Knowledge (K) | 0.13 | 0.15 | 0.22 | An(2)=0.32, p=0.74 |
| Comprenhension (U) | 0.12 | 0.00 | -0.12 | X2(2)=2.64, p=0.27 |
| Aplication (Ap) | 0.14 | 0.11 | 0.12 | X2(2)=0.08, p=0.96 |
| Syntethis (S) | 0.32 | 0.40 | 0.41 | An(2)=0.17, p=0.84 |
| **Total** | 0.21 | 0.21 | 0.20 | An(2)=0.01, p=0.99 |

**Table 5. Results of educational effectiveness for groups A, B and C**

| Level | G. Control 1 | G. Control 2 | G. Treatment | Stats |
|---|---|---|---|---|
| Knowledge (K) | 0.25 | 0.15 | 0.29 | An(2)=0.52, p=0.61 |
| Comprenhension (U) | 0.36 | 0.16 | -0.07 | An(2)=2.34, p=0.12 |
| Aplication (Ap) | 0.58 | 0.41 | 0.27 | An(2)=0.81, p=0.46 |
| Syntethis (S) | 0.60 | 0.69 | 0.56 | An(2)=0.20, p=0.82 |
| **Total** | 0.51 | 0.45 | 0.32 | An(2)=2.40, p=0.12 |

**Table 6. Results of educational effectiveness for groups C, D and E**

| Level | G. Control 1 | G. Control 2 | G. Treatment | Stats |
|---|---|---|---|---|
| Knowledge (K) | 0.40 | 0.13 | 0.27 | An(2)=2.40, p=0.17 |
| Comprenhension (U) | 0.67 | 0.33 | 0.00 | An(2)=1.47, p=0.32 |
| Aplication (Ap) | 0.95 | 0.74 | 0.85 | An(2)=2.85, p=0.15 |
| Syntethis (S) | 0.64 | 0.64 | 0.28 | An(2)=0.95, p=0.44 |
| **Total** | 0.71 | 0.56 | 0.33 | An(2)=5.03, p=0.05 |

**Table 7. Results of educational effectiveness for groups D and E**

### 3.4 Solutions to the practices

In this section we described the results obtained in the evaluation session. The results are shown using a scale from 0-1.

After checking the practices the results are displayed in Table 8. As can be observed the marks are very similar. In case of the control group 1 (ANTLR) and control group 2 (ANTLRworks) the marks are the same. There exists a small different in the treatment group (VAST) but it is not significant. D

| | |
|---|---|
| **Average ANTLR** | 0,85 |
| **Average ANTLRworks** | 0,85 |
| **Average VAST** | 0,88 |

**Table 8. Average of practice's marks (first revision)**

During the correction we detected that some students of the control group 1 had used visualization tools as JFlap and ANTLRworks. Besides, we performed a second correction changing some aspects. The results are shown in Table 9. In this case the marks of the three groups is the same.

| | |
|---|---|
| **Average ANTLR (no vis)** | 0,86 |
| **Average ANTLRworks** | 0,86 |
| **Average VAST** | 0,86 |

**Table 9. Average of practice's marks (second revision)**

## 4 Conclusions

The results obtained in this evaluation do not show any significant difference using or not visualization tools. This kind of tools clearly help to the learning task [4], however, there exists cases [5] where the results are not the expected. This can be due to different reasons. From the results we think that the knowledge test (pre and postest) and the tasks performed during the evaluation were not adequate given the features of VAST. As fundamental aspect, we plan to study how to use VAST to obtain the results in the learning task.

Thanks to the comments obtained in this evaluation, we will add new features to VAST, so we plan to:

- Improve the importing process, so it is necessary to save the last parser import and to show the result of the compilation process.
- Auto configuration of the tool. It is necessary that the tool detects automatically the virtual machine.
- Add tooltips.
- Improve the time of execution.
- Expand the grammar in the annotation process to make possible to work with grammars which use recursive rules at the axiom level.
- Improve the grammar editor.
- Export the animations.
- Help menu.

Implementing all this changes we plan a second global integration which adds more functionality to VAST. Besides, we plan a long term evaluation with the tool.

## Acknowledgment

This project is supported by project TIN2008- 04103/TSI of the Spanish Ministry of Science and Innovation.

## References


1. F. Almeida-Martínez, J. Urquiza-Fuentes, and J. Velázquez-Iturbide. Visualization of syntax trees for language processing courses. Journal of UniversalComputer Science, 5(7):1546{1561, 2009.
2. Jean Bovet and Terence Parr. ANTLRWorks: an antlr grammar development environment. Software: Practice and Experience, 38(12):1305–1332, 2008.
3. Terence Parr. The Definitive ANTLR Reference: Building Domain-Specific Languages. Pragmatic Bookshelf, 2007.
4. Thomas L. Naps, Guido Rossling, Vicki Almstrum, Wanda Dann, Rudolf Fleischer, Chris Hundhausen, Ari Korhonen, Lauri Malmi, Myles McNally, Susan Rodger, and J. Ángel Velázquez-Iturbide. Exploring the role of visualization and engagement in computer science education. In ITiCSE-WGR '02: Working group reports from ITiCSE on Innovation and Technology in Computer Science Education, pages 131–152, New York, NY, USA, 2002. ACM.
5. Colleen Kehoe, John T. Stasko, and Ashley Taylor. Rethinking the evaluation of algorithm animations as learning aids: an observational study. International Journal of Human-Computer Studies, 54(2):265–284, 2001.

## Appendix A: Pretest and postest of knowledge

1- What are the main features of the LL (1) parsers?

2- What is a derivation?

3- Given the following grammar:

S::= FS'
S'::=+FS' | -FS' |λ
F : = id | cte | ( S )

Show the cases of a non recursive LL (1) parser **pointing out the state of the input stream and the stack before and after** of each of the following operations:

a) A derivation
b) The recognition of a token

4- Using the same grammar of the previous exercise, for which it has been developed a **non recursive LL (1)** parser and the parser state: **(S ) S'$, id – cte + cte)-cte $)**. Detail the following steps of the parser until it reaches to the symbol S'

5- Design a LL (1) grammar to recognize logic expressions with the following characteristics:

- The operands are the logic constants *true* and *false*
- The operators are the *parenthesis*, the unit operator *not* and the binary operators *xor*, *and*, *or*, *nand* and *nor*. The associative of the binary operators is on the right. The precedence order is specified in the following table:

| | |
|---|---|
| ( ) | **+ Higher precedence** |
| `Not` | |
| `and nor` | Both the same precedence |
| `or nand` | Both the same precedence |
| `xor` | **- Lower precedente** |

- The precedente has to be implemented in the own grammar. If an operator O1 has higher precedence than another operator O2, then the consequents of O1 and its operands will be always processed before the operands of O2.
- Some examples of correct input streams are:

```
not true nor false and false
false or true nand not ( true xor not false )
```

## Appendix B: ANTLR's opinion questionnaire

In the questions of this test, mark a value using the scale of the following table. According to the question, it will refer to **opinion** or **quality**:

| Value | Opinion | Quality |
|---|---|---|
| 1 | Not agree | Very bad |
| 2 | A bit agree | Bad |
| 3 | Without opinion | Regular |
| 4 | Agree | Well |
| 5 | Very agree | Very well |

1. I think that ANTLR is **easy to use [ ]**
   The parts, which I consider **the most difficult** to use, are:

2. I think that ANTLR **has helped me** to understand the way of working of the LL (1) parsers, in particular in:
   **[ ]** The building process of the syntax tree
   **[ ]** The processing of the input stream
   **[ ]** The parser's stack

3. I think that **the general quality** of ANTLR to show the way of working of the LL (1) parsers is high: **[ ]**
   The parts with the **best quality** for you are:
   The parts with the **worst quality** for you are:

4. Give your opinion about the **ease of use** and the **quality** of the different aspects of ANTLR:

| | Easy to use | Quality |
|---|---|---|
| Grammar edition | | |
| Building of parser | | |
| Visualization of the analysis process | | |

5. In general, I like ANTLR: **[ ]**

Answer to the following questions in a free format:

6. Say what features could be useful but ANTLR does not implement:

7. Say what features of ANTLR you would remove:

8. Describe the **positive aspects** which you find in ANTLR:

9. Describe the **negative aspects** that you find in ANTLR:

## Appendix C: ANTLRworks' opinion questionnaire

In the questions of this test, mark a value using the scale of the following table. According to the question, it will refer to **opinion** or **quality**:

| Value | Opinion | Quality |
|---|---|---|
| 1 | Not agree | Very bad |
| 2 | A bit agree | Bad |
| 3 | Without opinion | Regular |
| 4 | Agree | Well |
| 5 | Very agree | Very well |

1. I think that ANTLRworks is **easy to use [ ]**
   The parts, which I consider **the most difficult** to use, are:

2. I think that ANTLRworks **has helped me** to understand the way of working of the LL (1) parsers, in particular in:
   **[ ]** The building process of the syntax tree
   **[ ]** The processing of the input stream
   **[ ]** The parser's stack

3. I think that **the general quality** of ANTLRworks to show the way of working of the LL (1) parsers is high: **[ ]**
   The parts with the **best quality** for you are:
   The parts with the **worst quality** for you are:

4. Give your opinion about the **ease of use** and the **quality** of the different aspects of ANTLRworks:

| | Easy to use | Quality |
|---|---|---|
| Grammar edition | | |
| Interpreter | | |
| View of the input stream | | |
| View of the syntax tree | | |
| Debugger | | |
| View of the input stream | | |
| View of the stack | | |
| View of events | | |

| | | |
|---|---|---|
| Debugger controls | | |

5. In general, I like ANTLRworks: **[ ]**

Answer to the following questions in a free format:

6. Say what features could be useful but ANTLRworks does not implement:
7. Say what features of ANTLRworks you would remove:
8. Describe the **positive aspects** which you find in ANTLRworks:
9. Describe the **negative aspects** that you find in ANTLRworks:

## Appendix D: VAST's opinion questionnaire

In the questions of this test, mark a value using the scale of the following table. According to the question, it will refer to **opinion** or **quality**:

| Value | Opinion | Quality |
|---|---|---|
| 1 | Not agree | Very bad |
| 2 | A bit agree | Bad |
| 3 | Without opinion | Regular |
| 4 | Agree | Well |
| 5 | Very agree | Very well |

1. I think that VAST is **easy to use [ ]**
   The parts, which I consider **the most difficult** to use, are:

2. I think that VAST **has helped me** to understand the way of working of the LL (1) parsers, in particular in:
   **[ ]** The building process of the syntax tree
   **[ ]** The processing of the input stream
   **[ ]** The parser's stack

3. I think that **the general quality** of VAST to show the way of working of the LL (1) parsers is high: **[ ]**
   The parts with the **best quality** for you are:
   The parts with the **worst quality** for you are:

4. Give your opinion about the **ease of use** and the **quality** of the different aspects of VAST:

| | Easy to use | Quality |
|---|---|---|
| Main menu | | |
| Icons | | |
| Animation controls | | |
| Glogal view | | |
| Subtrees | | |
| Zoom | | |
| View of the input stream | | |
| View of the stack | | |
| Configurations | | |
| Manage of configuration | | |
| Edit the input stream | | |
| Parser import | | |

5. In general, I like VAST: **[ ]**

Answer to the following questions in a free format:

6. Say what features could be useful but VAST does not implement:

7. Say what features of VAST you would remove:

8. Describe the **positive aspects** which you find in VAST:

9. Describe the **negative aspects** that you find in VAST:

## Appendix E: Tasks of the evaluation

Given the following grammar to represent arithmetic expressions:

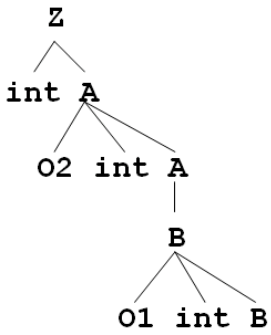
S := F N
N ::= + F N | -F N | * F N | / F N | λ
F ::= id | cte | ( S )

- **Build a visualization** where all the operators are used at least once: add, sub, mul, div and parenthesis.
- **Change the grammar** (if it is necessary) to make the operators have the following precedence:

| | |
|---|---|
| ( ) | **+ Higher precedence** |
| * / | Both the same precedente |
| – | |
| + | **- Lower precedence** |

Take into account that all the binary operators are associative on the right.

**NOTE**: An operand O1 has higher precedence than another operand O2 when all the consequents of O1 and the own operator have to be processed before the consequents of O2 and its operands.




- Document the result with various visualizations (1 or more). Create a Word document where it is shown one or more examples with the following elements:
  - the input stream
  - a screenshot (**`Alt+Impr. Pant`**.) where the precedence and associativies are shown.
  - a text explanation.

.