JGrail Specification
Aaron Cooper
University of Prince Edward Island
Contents

1 Introduction 3

2 What is Grail+? 3

3 What is JGrail? 3
   3.1 Software Environment ........................................... 3
   3.2 Obtaining JGrail ................................................. 3

4 JGrail Setup 3
   4.1 Extracting JGrail.tar ............................................. 3
      4.1.1 JGrail directory ........................................... 3
      4.1.2 .jgrail hidden directory .................................. 4
   4.2 Running JGrail .................................................. 4
   4.3 Compiling JGrail ................................................ 4
      4.3.1 Method 1 - Makefile ...................................... 4
      4.3.2 Method 2 - Java Compiler .................................. 4

5 Future Work & Known Issues 4
   5.1 Grail+ "fl" functions ........................................... 4
   5.2 Empty Spaces .................................................. 5
   5.3 Error Messages ................................................ 5
   5.4 Combo Box nuisance ............................................. 5
   5.5 Linux version .................................................. 5

6 Graphical Interface Specification 5
   6.1 JGrail Main Screen ............................................. 5
      6.1.1 File Menu ................................................ 5
      6.1.2 Edit Menu ............................................... 5
      6.1.3 Configure Menu .......................................... 5
      6.1.4 Help Menu ............................................... 5
   6.2 JGrail Configuration Window: Function Modifications .......... 6
      6.2.1 JGrail Configuration Window: Path Settings ............... 6

7 Appendix - Figures 7

List of Figures

1 JGrail Main Screen ........................................... 7
2 JGrail Configuration Window: Function Modifications (Unary functions) .... 8
3 JGrail Configuration Window: Function Modifications (Binary functions) ...... 9
4 JGrail Configuration Window: Path Settings ................................ 10
1 Introduction

During May 2004, a single-student project was conducted at the University of Prince Edward Island (UPEI) by a recent graduate under the supervision of Dr. Cezar Câmpeanu. The project involved the construction of a Graphical User Interface (GUI) to existing command-line programs. Among the project goals includes the ability to use the software as a teaching tool in the Theory of Computing course. This document details the technical aspects of the graphical user interface software entitled JGrail.

2 What is Grail+?

Grail+ is a set of command-line tools developed by researchers at the University of Western Ontario. These tools provide operations for regular expressions, finite automata, and finite languages. More information can be obtained at the Grail+ home page [http://www.csd.uwo.ca/research/grail/grail.html].

3 What is JGrail?

JGrail is a Java-based graphical user interface program. As opposed to the Grail+ command-line tools, JGrail provides the operations for regular expressions, finite automata, and finite languages in a graphical environment. As it currently stands, it is simply a front-end to the Grail+ toolset; however, its operations can be configured to use any back-end program.

3.1 Software Environment

The software was developed on Sun’s SunOS 5.8 operating system and used Sun’s Java 1.4.2_04 for compilation and execution. The 2.5 version of Grail+ was used as the back-end toolset, but it also works with Grail+ 3.0.

3.2 Obtaining JGrail

JGrail should be available as a tar file from Dr. Cezar Câmpeanu. For more information, please contact him at campeanu@upei.ca

4 JGrail Setup

4.1 Extracting JGrail.tar

To extract the files, type the following command:

```
tar -zxvf jgrail.tar
```

Upon extraction, two directories will be created: JGrail, which contains the java files, and jgrail, which is a hidden directory that stores configuration settings. Ensure that the jgrail hidden directory is placed under the user’s home folder (e.g., /home/user/jgrail), as JGrail will look in this location for its settings.

4.1.1 JGrail directory

The JGrail directory contains the java source and class files, under the subdirectories "source" and "classes" respectively. Under the "source" directory, there exists four subdirectories: Left, Menu, Right, and third_party.

The JGrail directory should appear as follows:

```
/home/user/JGrail
/home/user/JGrail/classes
/home/user/JGrail/source
```
4.1.2 jgrail hidden directory

The jgrail hidden directory stores configuration and help files. The configuration file can be manually edited from the command-line or via the GUI, as explained in section 6.0 - Graphical Interface Specification.

4.2 Running JGrail

As the java class files are included in the jgrail.tar archive, there is no need to compile JGrail in order to use it. To begin JGrail, assuming the archive was extracted to /home/user, type the following command:

java -classpath /home/user/JGrail/classes JGrailGui

4.3 Compiling JGrail

In the event that there is a need to compile the software, there are two options available. The first method uses a makefile, while the second invokes the java compiler directly. Both methods assume the directory structure created by the extraction process described earlier. The compilation should be performed under the JGrail/source directory.

4.3.1 Method 1 - Makefile

This method describes the compilation strategy used throughout the development of JGrail. In the event that problems arise, a more direct method described in Method 2 can be attempted.

To compile, use the makefile found in the "source" directory. Simply type the following command:

make -f Makefile

This step compiles all appropriate java source files and places the class files under the "JGrail/classes" directory.

4.3.2 Method 2 - Java Compiler

This method involves invoking the java compiler directly. In order to make the process easier, all the java source files should be placed under the same directory. Copy all the source java files from subdirectories Left, Right, Menu, and third_party into target folder "source".

With all the java source files contained in the "source" directory, invoke the java compiler via the following command:

javac -verbose -classpath ../../classes -d ../classes JGrailGui.java

As with the first method, this command will place the resulting class files under the "classes" directory.

5 Future Work & Known Issues

5.1 Grail+ "fl" functions

The finite language Grail+ functions do not run properly upon invocation. The programs would appear to simply hang, showing no signs of execution. The functions in question are suffixed by "fl", such as flexec and firevers.
5.2 Empty Spaces
When spaces are placed between characters in JGrail’s input boxes, the back-end Grail+ programs would disregard all characters after the space. For example, if the regular expression "a + b" was inputted, the corresponding Grail+ programs would read the input as "a" only.

5.3 Error Messages
In an attempt to reduce the number of dialog boxes presented to the user, error and status messages are displayed in the window’s status bar. This may not be the best manner to present this information. An alternative method would be to place this information in a separate window, which could be accessed via a menu on the toolbar. In this instance, the status bar could be used to signify that an error occurred, and that the user should refer to the error window for more detailed information.

5.4 Combo Box nuisance
The SteppedComboBox works fine on maximized screen, but when the window is shrunk, the combo box’s "popup" area does not grow larger than the size of the combo box itself. That is, when the user clicks the arrow to view the list, the list’s popup area is much smaller than the combo box. Therefore, it is possible that the list’s items are not entirely viewable.

5.5 Linux version
The Grail+ toolset is currently not available for a Linux environment. To support a larger user base, porting this toolset to Linux would be very beneficial.

6 Graphical Interface Specification
This section describes the graphical features of JGrail, as depicted in the screen shots at the end of the document.

6.1 JGrail Main Screen
The main system window where the majority of the user interactions occur and it is illustrated in Figure 1
The toolbars options are as follows:

6.1.1 File Menu
From the File Menu, the user can save their input and output to text files. That is, the information that is contained in the "First Input", "Second Input" and "Output" graphical boxes can be stored for future use. In addition, the user can exit the program from this menu.

6.1.2 Edit Menu
The Edit Menu contains the following three standard operations: Cut, Copy, and Paste.

6.1.3 Configure Menu
The Configure Menu provides an option that enables the user to modify some program settings. The JGrail configuration screen is further described in subsections 6.2 and 6.2.1 below.

6.1.4 Help Menu
The Help Menu contains a short list of questions and answers (similar to a FAQ), as well as version information about JGrail.
6.2 JGrail Configuration Window: Function Modifications

The JGrail Configuration Window contains three tabs. The first two tabs provide options to modify the back-end programs used for handling the unary and binary functions, while the third tab deals with the default locations of JGrail folders. Figure 2 displays the Unary Functions tab, which is identical to the Binary Functions tab, illustrated in Figure 3.

From this tab, the user can add new unary functions to JGrail, as well as delete existing ones. In addition, existing command names and their descriptions can be modified via clicking on the appropriate cell in the grid.

This screen demonstrates that JGrail can be configured to use back-end programs from several toolsets. Therefore, although JGrail currently uses the Grail+ tools, the software can easily be modified to utilize functions from other toolsets.

6.2.1 JGrail Configuration Window: Path Settings

This window enables the user to modify JGrail folder locations, see Figure 4. The Function Folder Location represents the folder that houses the back-end programs; in this instance, the location where the Grail+ executables are stored. The Save Folder Location represents the folder where JGrail stores the input and output text files saved by the user (via the File Menu).
7 Appendix - Figures

Figure 1: JGrail Main Screen
Figure 2: JGrail Configuration Window: Function Modifications (Unary functions)
<table>
<thead>
<tr>
<th><strong>Unary Functions</strong></th>
<th><strong>Binary Functions</strong></th>
<th><strong>Default Paths</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command</strong></td>
<td><strong>Description</strong></td>
<td></td>
</tr>
<tr>
<td>append</td>
<td>append two finite languages</td>
<td></td>
</tr>
<tr>
<td>inter</td>
<td>intersect a finite language with a finite-state machine</td>
<td></td>
</tr>
<tr>
<td>lq</td>
<td>left quotient</td>
<td></td>
</tr>
<tr>
<td>prod</td>
<td>product of two finite languages</td>
<td></td>
</tr>
<tr>
<td>product</td>
<td>cross product of two finite-state machines</td>
<td></td>
</tr>
<tr>
<td>union</td>
<td>union of two finite languages</td>
<td></td>
</tr>
<tr>
<td>union</td>
<td>union of two finite-state machines</td>
<td></td>
</tr>
<tr>
<td>regular</td>
<td>union of two regular expressions</td>
<td></td>
</tr>
<tr>
<td>concat</td>
<td>concatenate two machines</td>
<td></td>
</tr>
<tr>
<td>concat</td>
<td>concatenate two regular expressions</td>
<td></td>
</tr>
<tr>
<td>finite</td>
<td>test a word for membership in the language of a finite...</td>
<td></td>
</tr>
<tr>
<td>isomorph</td>
<td>test two finite-state machines for isomorphism</td>
<td></td>
</tr>
<tr>
<td>isomorph</td>
<td>test a word for membership in a finite language</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3:** JGrail Configuration Window: Function Modifications (Binary functions)
Figure 4: JGrail Configuration Window: Path Settings