setlX v2.5.3

Interpreter Manual

Herrmann, Tom

November 26, 2016

"Computers make very fast, very accurate mistakes."

Contents

1 Overview

This is the manual of SETLX, an interpreter for the very high level programming language SETLX (set language extended).

The most distinguishing feature of this language is the support it offers for sets and lists. As set theory is the language of mathematics, many mathematical algorithms that are formulated in terms of set theory have very straightforward implementations in SETLX.

Designed mostly by Karl Stroetmann, the SETLX language is an evolution of SETL by Jack Schwartz. It was specifically conceived to make the unique features of SETL more accessible to today's computer science students.

The SETLX interpreter, which currently is the SETLX reference implementation, was implemented by Tom Herrmann. Its official homepage is http://setlX.randoom.org/. You may send bug-reports and/or questions about the SETLX interpreter via e-mail to setlx@randoom.org, or create an issue on github (https://github.com/herrmanntom/setlX).

2 Installation

To comfortably use SETLX it may be installed into the search-path. After this is done, it may be launched just like any other command-line program on your system.

2.1 Unix-like OS (Linux, MacOS X, etc.)

1. Copy all '`setlX*.jar`'-files into some folder, which is accessible by all users, who should be able to execute SETLX (e.g. '`/usr/local/setlX/`').
   You may also copy them into the home directory of some user, when she is the only SETLX-user on the system (e.g. '`/home/<user>/programs/setlX/`').
2. Copy the contents of the default '`setlXlibrary`' into some folder for each user (e.g. '`/home/<user>/setlXlibrary/`').
3. Open the '`setlX`'-script in your favorite editor (either '`vi`' or '`emacs`' of course).
4. Change the contents of the '`setlXJarDirectory`'-variable to the path where you copied the jar-files, e.g.

   ```
   setlXJarDirectory="/usr/local/setlX/"
   ```

5. Change the '`setlXlibraryPath`'-variable to the path where SETLX should look for library files, e.g.

   ```
   setlXlibraryPath="$HOME/setlXlibrary/"
   ```

6. Save and close the file.
7. Copy the modified '`setlX`'-script into some folder, which is in the search-path of all users which should be able to execute SETLX (e.g. '`/usr/local/bin`').
   When only used by one user, you may copy it into the 'bin' directory in her home instead (e.g. '`/home/<user>/bin`').
8. Make the copied script executable, e.g.

```
chmod +x /usr/local/bin/setlX
```

2.2 Microsoft Windows

"Here's a nickel, kid. Go buy yourself a real computer."[1]

1. Copy all '`setlX*.jar`' files and the '`setlX.cmd`' file into some folder, which is accessible by all users, who should be able to execute SETLX (e.g. '`C:\Program Files\setlX\`').
2. Copy the contents of the default '`setlXlibrary`' into some folder for each user (e.g. '`C:\Program Files\setlX\library`').
3. Open the copied '`setlX.cmd`'-script in your favorite text editor.
4. Change the contents of the '`setlXJarDirectory`'-variable to the path where you copied the jar-files, e.g.

   ```
   set setlXJarDirectory=C:\Program Files\setlX\
   ```

5. Change the '`SETLX_LIBRARY_PATH`'-variable and set it to the path where SETLX should look for library files, e.g.

   ```
   set SETLX_LIBRARY_PATH=C:\Program Files\setlX\library
   ```

6. Save and close the file.
7. Add the folder you placed both files into to the search-path (requires Administrator privileges):
   a) Press `[Win]+[Pause]` key combination to open system preferences.
   b)
   - On Windows Vista or newer:
     Click on '`Advanced system settings`' on the left pane, which opens a new window.
   - On Windows XP or older:
     Click the '`Advanced`' tab in the current window.

   c) Click on '`Environment Variables`' on the bottom, which opens another window.
   d) Select the '`Path`' variable in the '`System variables`' section and click on '`Edit...`', which opens jet another window.
   e) Do not alter the existing content of the '`Variable value`' field, but add a semicolon ('`;`'), followed by the full path to the folder where you installed setlX, to its very end. Using the previous example path, the new value should look like:

      ```
      <previous value>;C:\Program Files\setlX
      ```

   f) Accept all changes by clicking '`OK`' in all windows you opened before.
   g) End you current session and login into Windows again.

[1] In reference to http://dilbert.com/1995-06-24

3 Usage

The SETLX interpreter has two basic modes of operation:

- file execution mode, which can be launched with

  ```
  setlX <path>/<name>.stlx
  ```

- interactive mode, which can be launched with

  ```
  setlX
  ```

The 'file execution mode' will be started if one or more paths to code files are supplied as parameters for this program. These files will then be parsed and executed in the order in which they are listed as parameters. When executing multiple files, they share the outermost scope.

When no paths are supplied, the 'interactive mode' will be started.
Executing the '`exit;`' statement will terminate the interpreter.

Additional parameters

In addition to file-paths a number of options can be used when running SETLX:

`-l <path>, --libraryPath <path>`
: Override SETLX_LIBRARY_PATH environment variable.

`-a, --noAssert`
: Disables all assert functions.

`-n, --noExecution`
: Load and check code for syntax errors, but do not execute it (has no effect in interactive mode).

`-r, --predictableRandom`
: Always returns the same pseudo random sequence of numbers from the internal random number generator (for debugging).

`-p <argument> ..., --params <argument> ...`
: Pass all following arguments to the executed program via 'params' variable.

`-e <expression>, --eval <expression>`
: Evaluates next argument as expression and exits (cannot be combined with execution of a statement or files).

`-x <statement>, --exec <statement>`
: Executes next argument as statement and exits (cannot be combined with evaluation of an expression or execution of files).

`-v, --verbose`
: Display the parsed program before executing it (has no effect in interactive mode).

`--doubleDefault`
`--doubleScientific`
`--doubleEngineering`
`--doublePlain`
: Sets how the exponent of doubles is printed.

`-h, --harshWelcome`
: Interactive mode: Reduce welcome message to a bare minimum.

`-m, --multiLineMode`
: Interactive mode: Only accept input after additional new line.

`--help`
: Displays some helpful information.

`--version`
: Display interpreter version and terminate.

4 System Requirements

To run SETLX a Java runtime (JRE) is required, which is compatible to Java version 1.7 (aka version 7) or higher.

When building from source, the Java Development Kit (JDK) version 1.7 (but not higher) and an Apache Maven installation have to be present as well.

The hardware requirements are highly dependent on the executed SETLX program. At least 512 MB of free main-memory should be available for optimal performance in most situations.

5 Limitations

5.1 Out of Memory Error

Due to its implementation and execution by the Java Virtual Machine (JVM), an 'out of memory error' might be encountered when executing SETLX programs using deep recursion, very large or very many sets and/or lists.

The JVM is only able to dynamically increase its memory allocation in very limited bounds, no matter how much free memory is available.

To work around these problems, edit the used launching script ('`setlX`' or '`setlX.cmd`') with a text editor and follow the directions given at the top.

5.2 Input in Interactive Mode on Unix

On UNIX (-like) systems, the prompt in interactive mode does not handle control sequences correctly. This is a Java limitation and no platform independent solutions are available. You may install the 'rlwrap' program to work around this issue.

6 Building from source

Optionally, when the interpreter source is available, the interpreter can be (re)build by executing

```
mvn clean install
```

in the '`interpreter`' folder.

A self-contained '.jar' file will be created, which can be launched on all Java-Platforms without additional scripts, jars or environment-variables by executing

```
java -jar setlX.jar <path>/<name>.stlx
```

or

```
java -jar setlX.jar
```

However, the `setlX` and `setlX.cmd` scripts can also be used to launch the jar file created by this process.

7 Disclaimer

```
You may distribute setlX under the terms of the BSD license,
which is stated at the end of this document.

This program uses the ANTLR parser generator,
which is also licensed under the BSD license.

The functions for plotting require jfreechart,
which is also licensed under the LGPL licence.

[The "BSD license"]
Copyright (c) 2011-2015, Tom Herrmann
All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the following
conditions are met:

 1. Redistributions of source code must retain the above
    copyright notice, this list of conditions and the following
    disclaimer.
 2. Redistributions in binary form must reproduce the above
    copyright notice, this list of conditions and the following
    disclaimer in the documentation and/or other materials
    provided with the distribution.
 3. The name of the author may not be used to endorse or
    promote products derived from this software without
    specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.
```

Figure 1: setlX license