The luamodulartables and luaset \LaTeX{} packages

Chetan Shirore, Ajit Kumar

Abstract

The luamodulartables package was developed by us to generate modular addition and multiplication tables for positive integers, for use in \LaTeX{} documents. The commands in the package have optional arguments for formatting of tables. These commands can be used in an environment similar to the tabular and array environments. The commands can also be used with the booktabs package to provide better formatting of tables in \LaTeX{}.

Similarly, the luaset package is developed by us to define finite sets and perform different operations on them inside \LaTeX{} documents. There is no special environment in the package for performing set operations. The package commands can be used in any environment (including mathematics environment).

These packages are written in Lua, and the \LaTeX{} source is to be compiled with the Lua\LaTeX{} engine. There is no need to install Lua on users’ systems as \LaTeX{} distributions (\TeX{} Live and MiKTeX) come bundled with Lua\LaTeX{}. The packages can be modified or extended by writing custom Lua programs.

1 Introduction

The Lua [1] programming language is a scripting language which can be embedded across platforms. With Lua\LaTeX{} [3], and more easily with the luacode [2] package, it is possible to use Lua in \LaTeX{}. The \LaTeX{} [9] and \LaTeX{} languages provide for programming [8]. However, with the internals of \LaTeX{} there are several limitations, especially for performing calculations on numbers in \LaTeX{} documents. There are packages like pgf [7] and xparse [10] in \LaTeX{} which provides some programming capabilities inside \LaTeX{} documents. However, such packages do not provide the complete programming structure that general programming languages, such as Lua, provide. The luacode [2] package is used in development, in addition to the xkeyval package.

The modular addition (multiplication) of integers with respect to a positive integer \(n\) is obtained by taking the remainder of the usual addition (multiplication) after dividing it by \(n\). There is no easy way in \LaTeX{} to do modular addition and multiplication [4]. With Lua, it can be achieved easily in \LaTeX{}. Also, non-Lua ways of doing modular arithmetic in \LaTeX{} are more complicated [5].

The time required for the Lua\LaTeX{} compilation to generate modulo addition and multiplication tables with the luamodulartables package, or to perform different operations on sets with the luaset package, is not an issue.

2 Installation and license

The installation of luamodulartables and luaset package is similar to simple \LaTeX{} packages, with a .sty file in the \LaTeX{} directory of a \TeX{}mf tree. The packages can be loaded with \texttt{\usepackage{luaset}} and \texttt{\usepackage{luamodulartables}} commands in the preamble of a \LaTeX{} document. The \TeX{} file is to be compiled using the Lua\LaTeX{} engine.

luamodulartables and luaset packages are released under the \LaTeX{} Project Public License v1.3c or later. The complete license text is available at \url{latex-project.org/lppl.txt}. The packages are developed in Lua. Lua is available as certified open source software. Its license is simple and liberal, compatible with the GNU GPL. A small part of the development of these packages was inspired by questions on \url{https://tex.stackexchange.com}. The content on this site is available under the CC BY-SA license.

3 The luamodulartables package

\texttt{\textbackslash luaModularMult} and \texttt{\textbackslash luaModularAdd} are the two basic commands in the luamodulartables package, to generate modular multiplication and addition tables, respectively. The command \texttt{\luaModularMult} has the following syntax and it is used to generate modular multiplication tables for positive integers.

\begin{verbatim}
\luaModularMult
\[multlabel=(text),\]
\[headline=(text),midline=(text)\]
\{(n)\}
\end{verbatim}

The command has one compulsory argument \(\langle n \rangle\), and three optional arguments \texttt{multlabel}, \texttt{headline}, and \texttt{midline}. The compulsory argument denotes the positive integer \(n\) with respect to which modular multiplication is to be carried out.

The \texttt{multlabel} string denotes the label to be printed as the entry in the first row and first column of the generated \texttt{tabular} environment. Its default value is \$\times\$. The \texttt{headline} refers to the style of horizontal line after first row in \texttt{tabular} or \texttt{table} environment. The \texttt{midline} refers to the style of horizontal lines after second row till the second last row. The \texttt{headline} and \texttt{midline} strings are both empty by default.

The formatting of the top line (before the beginning of the first row) and the bottom line (after the end of the last row) are defined in the user’s \LaTeX{} document. The alignment of columns and use of vertical lines for columns are likewise specified in the document.

The luamodulartables and luaset \LaTeX{} packages
Table 1: Illustration of \texttt{\textbackslash luaModularMult}

<table>
<thead>
<tr>
<th>$\mathbb{Z}_4$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

An example of using the \texttt{\textbackslash luaModularMult} command follows, specifying the optional arguments \texttt{multlabel} and \texttt{headline}. It requires the \texttt{amsmath} and \texttt{amssymb} packages.

\begin{verbatim}
\begin{tabular}{r|rrrr}
\texttt{\textbackslash luaModularMult}[\texttt{multlabel}=$\mathbb{Z}_4$,
\texttt{headline}=4$] \ \% \\
{4} \ % \\
\end{tabular}
\end{verbatim}

This generates the output shown in Table 1.

Similarly, the command \texttt{\textbackslash luaModularAdd} is used to generate addition modulo tables for positive integers. It has the following syntax:

\begin{verbatim}
\texttt{\textbackslash luaModularAdd}[\texttt{addlabel}=\langle \texttt{text} \rangle,
\texttt{headline}=\langle \texttt{\textbackslash text} \rangle,\texttt{midline}=\langle \texttt{\textbackslash text} \rangle\] {\langle \texttt{n} \rangle}
\end{verbatim}

The \texttt{addlabel} argument denotes the label to be printed as the entry in the first row and the first column of tabular environment. Its default value is \texttt{+}. The optional parameters \texttt{headline} and \texttt{midline} are exactly the same as in the \texttt{\textbackslash luaModularMult} command.

4 The \texttt{luaset} package

4.1 Defining and displaying sets

A set can be defined with the \texttt{\textbackslash luaSetNew} command:

\begin{verbatim}
\texttt{\textbackslash luaSetNew}{\langle \texttt{name} \rangle}\{{\langle \texttt{set} \rangle}\}
\end{verbatim}

For example, the following defines sets \texttt{A} and \texttt{B}:

\begin{verbatim}
\texttt{\textbackslash luaSetNew}\{\texttt{A}\}\{a,b,c,10,d,10,a,30\}
\texttt{\textbackslash luaSetNew}\{\texttt{B}\}\{d,e,f,10,20\}
\end{verbatim}

The set can be output with \texttt{\textbackslash luaSetPrint}:

\begin{verbatim}
\texttt{\textbackslash luaSetPrint}\{\langle \texttt{name of set} \rangle\}
\end{verbatim}

Continuing our example, the commands

\begin{verbatim}
\langle \texttt{A} = \texttt{\textbackslash luaSetPrint}\{\texttt{A}\}\rangle \ %
\langle \texttt{B} = \texttt{\textbackslash luaSetPrint}\{\texttt{B}\}\rangle \%
\end{verbatim}

generates this output (notice that duplicate elements have been removed, this being a set):

\begin{verbatim}
A = \{10, 30, a, b, c, d\}
B = \{10, 20, d, e, f\}
\end{verbatim}

References

[1] Lua 5.4 reference manual. lua.org/manual/5.4
[2] Luacode package page. ctan.org/pkg/luacode
[3] LuaTeX package page. ctan.org/pkg/luatex
[4] Modular arithmetic in LaTeX. mathoverflow.net/questions/18813
[5] Modular arithmetic in LaTeX. tex.stackexchange.com/questions/34424
[7] PGF package page. ctan.org/pkg/pgf
[10] \texttt{xparse} package page. ctan.org/pkg/xparse

Chetan Shirore, Ajit Kumar

The command \texttt{\textbackslash luaSetPrint} parses, sorts and prints all elements by using the \texttt{parsers.iterator} [6] function in lualibs.

4.2 Commands in the \texttt{luaset} package

These commands are available in the \texttt{luaset} package to perform various operations on finite sets in \LaTeX\ documents.

\begin{verbatim}
\texttt{\textbackslash luaSetUnion}\{\texttt{C}\}\{\texttt{A}\}\{\texttt{B}\} \% Defines new set \texttt{C} as union of sets \texttt{A} and \texttt{B}.
\texttt{\textbackslash luaSetIntersection}\{\texttt{C}\}\{\texttt{A}\}\{\texttt{B}\} \% Defines new set \texttt{C} as intersection of sets \texttt{A} and \texttt{B}.
\texttt{\textbackslash luaSetDifference}\{\texttt{C}\}\{\texttt{A}\}\{\texttt{B}\} \% Defines new set \texttt{C} as difference of sets \texttt{A} and \texttt{B}.
\texttt{\textbackslash luaSetCardinal}\{\texttt{A}\} \% Gives cardinality of set \texttt{A}.
\texttt{\textbackslash luaSetBelongsTo}\{\texttt{x}\}\{\texttt{A}\} \% Returns true if \texttt{x} is in set \texttt{A}, otherwise returns false.
\texttt{\textbackslash luaSetSubset}\{\texttt{A}\}\{\texttt{B}\} \% Returns true if set \texttt{A} is a subset of set \texttt{B}, otherwise returns false.
\texttt{\textbackslash luaSetProper}\{\texttt{A}\}\{\texttt{B}\} \% Returns true if set \texttt{A} is a proper subset of set \texttt{B}, otherwise returns false.
\texttt{\textbackslash luaSetEqual}\{\texttt{A}\}\{\texttt{B}\} \% Returns true if set \texttt{A} is equal to set \texttt{B}, otherwise returns false.
\end{verbatim}

Chetan Shirore
Department of Mathematics,
Institute of Chemical Technology, Mumbai - 400019,
Maharashtra, India

Ajit Kumar
Department of Mathematics,
Institute of Chemical Technology, Mumbai - 400019,
Maharashtra, India