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 TeX source is to be compiled with the LuaLaTeX engine. There is no need to install Lua on users' systems as TeX distributions (TeX Live and MiKTeX) come bundled with LuaLaTeX. 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 LuaTeX [3], and more easily with the `luacode` [2] package, it is possible to use Lua in LaTeX. The TeX [9] and LaTeX languages provide for programming [8]. However, with the internals of TeX 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 LuaTeX 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 `texmf` tree. The packages can be loaded with `\usepackage{luaset}` and `\usepackage{luamodulartables}` commands in the preamble of a LaTeX document. The TeX file is to be compiled using the LuaLaTeX engine.

`luamodulartables` and `luaset` packages are released under the LaTeX Project Public License v1.3c or later. The complete license text is available at `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 `https://tex.stackexchange.com`. The content on this site is available under the CC BY-SA license.

3 The `luamodulartables` package

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

```
\luaModularMult
  [multlabel=⟨text⟩,
   headline=⟨text⟩,midline=⟨text⟩]
  {⟨n⟩}
```

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

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

doi.org/10.47397/tb/43-3/tb135shirore-luamath

$\mathbb{Z}_4$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Table 1: Illustration of `\luaModularMult`

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

```
\begin{tabular}{r|rrrr} \toprule
\luaModularMult[multlabel=$\mathbb{Z}_4$,
                headline=\midrule]
               {4} \\
\bottomrule \end{tabular}
```

This generates the output shown in Table 1.

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

```
\luaModularMult
  [addlabel=⟨text⟩,
   headline=⟨text⟩,midline=⟨text⟩]
  {⟨n⟩}
```

The `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 `$+$`. The optional parameters `headline` and `midline` are exactly the same as in the `\luaModularMult` command.

4 The `luaset` package

4.1 Defining and displaying sets

A set can be defined with the `\luaSetNew` command:

`\luaSetNew{`⟨*name*⟩`}{`⟨*set*⟩`}`

For example, the following defines sets A and B:

```
\luaSetNew{A}{a,b,c,10,d,10,a,30}
\luaSetNew{B}{d,e,f,10,20}
```

The set can be output with `\luaSetPrint`:

`\luaSetPrint{`⟨*name of set*⟩`}`

Continuing our example, the commands

```
\(A = \luaSetPrint{A}\) \\
\(B = \luaSetPrint{B}\)
```

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

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

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

4.2 Commands in the `luaset` package

These commands are available in the `luaset` package to perform various operations on finite sets in LaTeX documents.

`\luaSetUnion{C}{A}{B}` Defines new set C as union of sets A and B.

`\luaSetIntersection{C}{A}{B}` Defines new set C as intersection of sets A and B.

`\luaSetDifference{C}{A}{B}` Defines new set C as difference of sets A and B.

`\luaSetCardinal{A}` Gives cardinality of set A.

`\luaSetBelongsto{x}{A}` Returns true if x is in set A, otherwise returns false.

`\luaSetSubseteq{A}{B}` Returns true if set A is a subset of set B, otherwise returns false.

`\luaSetSubset{A}{B}` Returns true if set A is a proper subset of set B, otherwise returns false.

`\luaSetEqual{A}{B}` Returns true if set A is equal to set B, otherwise returns false.

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`

[6] parsers.iterator function in lualibs. `github.com/latex3/lualibs/blob/main/lualibs-util-prs.lua`

[7] PGF package page. `ctan.org/pkg/pgf`

[8] W.M. Richter. TeX and scripting languages. *TUGboat* 25(1):71–88, 2004. `tug.org/TUGboat/tb25-1/richter.pdf`

[9] Wikipedia. TeX. `en.wikipedia.org/wiki/TeX`

[10] Xparse package page. `ctan.org/pkg/xparse`

◇ 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