The `commandoptionals` package

Sergio Martinez-Losa

February 1, 2026

Abstract

The `commandoptionals` package extends standard LaTeX command definitions. It allows users to define commands with a flexible number of optional and mandatory arguments, checking that the total number does not exceed 9.

1 Introduction

Standard LaTeX `\newcommand` allows for at most one optional argument. The `xparse` package provides powerful tools for defining complex command interfaces. This package, built on top of `xparse`, provides a simplified interface `\newcommandoptionals` to define commands with multiple optional arguments and their default values.

2 Usage

2.1 Loading the Package

```
\usepackage{commandoptionals}
```

2.2 Defining Commands

The main command is `\newcommandoptionals`.

```
\newcommandoptionals{\<cmd>}[<num_opt>][<num_mand>][<def_1>]...[<def_n>]{<body>}
```

- `<cmd>`: The command to be defined.
- `<num_opt>`: Number of optional arguments (default 0).
- `<num_mand>`: Number of mandatory arguments (default 0).
- `<def_i>`: Default value for the i-th optional argument. You must provide as many default values as optional arguments specified.
- `<body>`: The definition of the command. Arguments are accessed via `#1`, `#2`, etc. Optional arguments come first, followed by mandatory arguments.

Constraint: <num_opt> + <num_mand> ≤ 9.

It is possible to have zero optional arguments (<num_opt> $= 0$) or zero mandatory arguments (<num_mand> $= 0$), provided the total sum is at least 0 (an empty command is possible, though perhaps not very useful).

3 Examples

3.1 Two Optional, One Mandatory

```
\newcommandoptionals{\mycmd}[2][1][DefaultA][DefaultB]{%
  Opt1: #1, Opt2: #2, Mand: #3%
}

\mycmd{Hello}          % -> Opt1: DefaultA, Opt2: DefaultB, Mand: Hello
\mycmd[UserA]{Hello}   % -> Opt1: UserA,    Opt2: DefaultB, Mand: Hello
\mycmd[UserA][UserB]{Hello} % -> Opt1: UserA, Opt2: UserB, Mand: Hello
```