typeset

Automatic typesetting framework for common GAP objects, with LaTeX generation

1.0

11 November 2022

Zachariah Newbery

Zachariah Newbery
Email: zn6@st-andrews.ac.uk
Homepage: https://zachnewbery.com

Contents

Chapter 1

Introduction

typeset is a package that implements an operation Typeset that can generate LaTeX string representations of a commonly used subset of mathematical objects within the GAP system.

Typeset is also built to be incredibly extensible, and can be easily extended to also support the generation of strings for other mark-up languages.

1.1 Core Functions

1.1.1 InfoTypeset

▷ `InfoTypeset` (info class)

Info class for the typeset package. Set this to the following levels for different levels of information:

- 0 - No messages
- 1 - Problems only: messages describing what went wrong, with no messages if an operation is successful
- 2 - Required preamble packages: displays informations about any required LaTeX packages that need to be added to the preamble to be rendered.
- 3 - Progress: also shows step-by-step progress of operations

Set this using, for example `SetInfoLevel(InfoTypeset, 1)`. Default value is 2.

1.1.2 Typeset

▷ `Typeset(`*`obj[, options]`*`)` (function)

Returns: String representation of object in given mark-up language, if `ReturnStr` option is set to `true`.

Typeset takes a mathematical object and generates a mark-up string representing that object in the given mark-up language. GAP options can also be added to modify the result:

- `ReturnStr` : Whether the method should return a string (`true`), or simply print the result (`false`). (default - `false`)

- `LDelim` : Left Delimiter for matrices. (default - `"("`)
- `RDelim` : Right Delimiter for matrices. (default - `")"`)
- `Lang` : Markup language of output, currently only `"latex"` is supported. (default - `"latex"`)
- `DigraphOut` : Typesetting method for Digraphs, one of `"dot"` to use raw dot within TeX, or `"dot2tex"` to convert the dot to native TeX. (default - `"dot"`)
- `SubCallOpts` : Alternate options for nested structures, passed as a record with the same options as the parent (but different values), or `false` if options are to stay the same between sub-calls. (default - `false`) either by specifying each options as an individual GAP options like below:

Example

```
gap> Typeset([[1, 2], [2, 1]] : LDelim := "[", ReturnStr := true);
"\\left[\\begin{array}{rr}\n1 & 2 \\\\\n2 & 1 \\\\\n\\end{array}\\right)\n"
```

or wrapping them in a record under an `options` GAP option, like:

Example

```
gap> Typeset([[1, 2], [2, 1]] : options := rec(LDelim := "[", ReturnStr := true));
"\\left[\\begin{array}{rr}\n1 & 2 \\\\\n2 & 1 \\\\\n\\end{array}\\right)\n"
```

or even simply passing a record object as the optional second argument:

Example

```
gap> Typeset([[1, 2], [2, 1]], rec(LDelim := "[", ReturnStr := true));
"\\left[\\begin{array}{rr}\n1 & 2 \\\\\n2 & 1 \\\\\n\\end{array}\\right)\n"
```

1.1.3 TypesetInternal

▷ `TypesetInternal(`*obj*`)` (function)

Returns: Typesetable String representation of object.

Generates a string representation of a passed GAP object that can be rendered by a typesetter. Called from the top-level method `Typeset` (1.1.2), which also passes a constructed options record as the `options` GAP option.

1.2 Core Operations

1.2.1 GenArgs (for IsObject)

▷ `GenArgs(`*obj*`)` (operation)

Returns: List of strings that describe the semantic structure of an object.

Generates the arguments describing the semantic definition of the passed mathematical object. This should result in a list that can be used to populate a format string in any mark-up language.

1.3 Constants and Utility Functions

1.3.1 MergeSubOptions

▷ `MergeSubOptions(`*`currOptions`*`)` (function)

Returns: Record representing the options that should be passed to any sub-calls within the current method.

Merges the passed options records passed optionally within a function call to `GenArgs` (**typeset: GenArgs for IsObject**). It allows for the `SubCallOpts` record value to specify any option values that should be altered on nested sub-calls when generating typesetting strings.

1.3.2 DEFAULT_TYPESET_OPTIONS

▷ `DEFAULT_TYPESET_OPTIONS` (global variable)

Default options record passed to Typeset. Merged with user-provided options to ensure correct construction of options for sub-calls, whilst also allowing option-less calls to the method.

Index