pattern-writing 0.2

Geraldo Xexéo

Tuesday 26th October, 2021 - 22:42

Contents

1 Introduction

This is a LaTeXstyle to help writing pattern languages [Wik].

It directly supports two formats for describing patterns: Portland style (if-then) and Coplien's style.

2 Options

`index`
`noindex`

These options turn pattern indexing true or false

`graph`
`nograph`

These option turn graph generation true or false

3 Commands

In this package you can:

- declare a pattern;
- describe a pattern in two formats, Portland and Coplien's;
- refer to a pattern;
- declara an anti-pattern;
- refer to an anti-pattern;
- create a graph, and
- configure some parameters.

3.1 Declaring patterns and anti-patterns

`pattern`
`anti-pattern`

`\pattern{`⟨*pattern-name*⟩`}`

`\antipattern{`⟨*anti-pattern-name*⟩`}`

Those macros declare a pattern or a anti-pattern and put the on the index if `index` is turned on. A pattern is also put in the graph, if `graph` is turned on.

3.2 Refering to patterns and anti-patterns

`patternref`
`antipatternref`

`\pattern[`⟨*how-to-print*⟩`]`pattern-name

`\antipattern[`⟨*how-to-print*⟩`]`antipattern-name

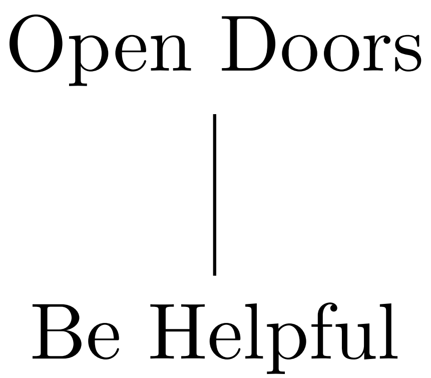

Figure 1: A figure

Those commands make a reference (with link) to patterns and anti-patterns, allowing for changes in the name, such as putting it in the plural.

3.3 Graph Commands

`\pstartgraph`
`\pstopgraph`
`\pgetgraph`
`\setPatternGraphLayout`

3.4 Pattern description

`\portland`

`\portland{`⟨*if part*⟩`}{`⟨*then part*⟩`}`

`\coplien`

`\coplien {`⟨*problem*⟩`}{`⟨*context*⟩`}{`⟨*forces*⟩`}{`⟨*solution*⟩`}`
`{`⟨*reasoning*⟩`}{`⟨*resultingcontext*⟩`}`

4 Example

4.1 Open Doors

IF there is a need to help students **THEN** is good to have a open-door police, and **Be Helpful** .

4.2 Be Helpful

IF there is a need to help people **THEN** you need to be really helpful

5 Code

5.1 Option Processing

index
noindex
graph
nograph

The style accepts two pair of options.

`index`/`noindex` controls if a index will be generated

`graph`/`nograph` controls if it will be possible to generate and use a graph

The default is `index` and `graph`

```
\newif\if@showindex\@showindextrue%
\newif\if@showgraph\@showgraphtrue%
\newif\if@graphstarted\@graphstartedfalse%
%
\DeclareOption{index}{\@showindextrue}%
\DeclareOption{noindex}{\@showindexfalse}%
\DeclareOption{graph}{\@showgraphtrue}%
\DeclareOption{nograph}{\@showgraphfalse}%
%
\ProcessOptions\relax%
```

5.2 Required Packages

`pattern-writing` requires xparse to use NewDocumentCommand and other syntax, requires `xcolor` to color anti-patterns, `makeidx` to control the index of patterns and `tikz` and sub-packages to draw the graph

```
\RequirePackage{xparse}%
\RequirePackage{xcolor}%
\if@showindex%
\RequirePackage{makeidx}%
\fi%
\if@showgraph%
\RequirePackage{pgf,tikz}%
\usetikzlibrary {graphs}%
%%\usetikzlibrary {graphs.standard}
\usetikzlibrary{graphdrawing}%
\usegdlibrary{circular,trees,force,layered}%
\fi%
%
```

5.3 Graph Support

With we use graph, some settings muste be enalbe

```
\if@showgraph%
\def\p@filename{graph.tikz}%
```

```
\def\p@CurrentPattern{ZERO}%
%
% standard layout for the graph
%
\def\p@GraphLayout{spring layout, node distance = 80mm}%
\NewDocumentCommand{\psetfilename}{m}{%
\def\p@filename{#1}%
}%
```

`\pstartgraph` This macro starts the processing of declarations to create the graph https://tex.stackexchange.com/questions/115932/on-the-basics-of-writing-to-reading-from-auxiliary-files-aux-toc-etc

```
%
\def\pw@gwidth{\textwidth}%
\def\pw@glen{!}%
%
\NewDocumentCommand{\pgraphwidth}{m}{%
\def\pw@gwidth{#1}%
}%
%
\NewDocumentCommand{\pgraphlen}{m}{%
\def\pw@glen{#1}%
}%
%
\NewDocumentCommand{\pstartgraph}{}{%
\newwrite\p@fileh%
\immediate\openout\p@fileh=\p@filename%
% tryed tikz, problem with accents
\immediate\write\p@fileh{\unexpanded{\resizebox}%
\@charlb\pw@gwidth\@charrb%
\@charlb\pw@glen\@charrb%
\@charlb%
\unexpanded{\begin{tikzpicture}  \graph []\p@GraphLayout%
\unexpanded{]} \@charlb}%
\@graphstartedtrue%
}%
```

`\pstopgraph` This macro stops the processing of declarations to create the graph https://tex.stackexchange.com/questions/115932/on-the-basics-of-writing-to-reading-from-auxiliary-files-aux-toc-etc

```
\NewDocumentCommand{\pstopgraph}{}{%
\immediate\write\p@fileh{\@charrb%
\unexpanded{;\end{tikzpicture}}\@charrb}%
\immediate\closeout\p@fileh%
\@graphstartedfalse%
}%
```

`\setPatternGraphLayout` This macro is used to set tikz commands for the graph layout. Please check tikz for possible values.

```
\NewDocumentCommand{\setPatternGraphLayout}{m}{%
    \def\p@GraphLayout{#1}%
}%
```

\pgetgraph

```
\NewDocumentCommand{\pgetgraph}{}{%
    \IfFileExists{\p@filename}%
    {\input{\p@filename}}%
    {}%
}%
```

```
\NewDocumentCommand{\pnode}{m}{%
    \if@graphstarted
    \immediate\write\p@fileh{"#1";}%
    \def\p@CurrentPattern{#1}%
    \fi
}%
\NewDocumentCommand{\pedge}{om}{%
    \if@graphstarted
    \IfNoValueTF{#1}
    {\immediate\write\p@fileh{"\p@CurrentPattern" -- "#2";}}%
    {\immediate\write\p@fileh{"#1" -- "#2";}}%
    \fi
}%
\else  % Nada
\def\p@filename{graph.tikz}%
\def\p@currentPattern{ZERO}%
\NewDocumentCommand{\psetfilename}{m}{}%
\NewDocumentCommand{\pstartgraph}{}{}%
\NewDocumentCommand{\pstopgraph}{}{}%
\NewDocumentCommand{\pgetgraph}{}{\includegraphics[width=\textwidth]{example-image-a}}%
\NewDocumentCommand{\psetcurrentpattern}{m}{%
}%
\NewDocumentCommand{\pgetcurrentpattern}{}{%
}%
\NewDocumentCommand{\pnode}{m}{}%
\NewDocumentCommand{\pedge}{om}{}%
\NewDocumentCommand{\setPatternGraphLayout}{m}{}%
\fi
```

5.4 Patterns

\pattern This macro declares a pattern (as a subsection).

```
\NewDocumentCommand{\pattern}{m}{\subsection{#1}\label{sec:#1}%
```

```
    \if@showindex%
    \index{#1|textbf}%
    \fi%
    \if@showgraph%
    \pnode{#1}
    \fi
}%
```

\patternref This macro makes a reference to a pattern, possibly changing the text in its option

```
\NewDocumentCommand{\patternref}{om}{%
    \IfNoValueTF{#1}%
    {\hyperref[sec:#2]{\textbf{#2}}}%
    {\hyperref[sec:#2]{\textbf{#1}}}%
    \if@showindex%
    \index{#2}%
    \fi%
    \if@showgraph%
    \pedge{#2}
    \fi
}%
```

5.5 Anti-Patterns

\antipattern This macro declares anti-pattern (as red text).

```
\NewDocumentCommand{\antipattern}{m}{\textcolor{red}{#1}%
    \label{sec:#1}%
    \if@showindex%
    \index{\textcolor{red}{#1}|textbf}%
    \fi%
}%
```

\antipatternref This macro makes a reference to an anti-pattern, also in red text

```
\NewDocumentCommand{\antipatternref}{om}{%
    \IfNoValueTF{#1}%
    {\hyperref[sec:#2]{\textcolor{red}{\textbf{#2}}}}%
    {\hyperref[sec:#2]{\textcolor{red}{\textbf{#1}}}}%
    \if@showindex%
    \index{\textcolor{red}{#2}}%
    \fi%
}%
```

5.6 Pattern Writing Styles

`\setportlandkeys` \setportlandkeys{⟨*if-key*⟩}{⟨*then-key*⟩} sets the values for if-then when writing a Portland-style pattern

```
\NewDocumentCommand{\setportlandkeys}{mm}{%
    \def\portland@ifkey{#1}%
    \def\portland@thenkey{#2}%
}%
```

`portland` \portland{⟨*if part*⟩}{⟨*then part*⟩}

```
\setportlandkeys{SE}{ENTÃO}
\NewDocumentCommand{\portland}{mm}{%
\textbf{\portland@ifkey~} #1 \textbf{~\portland@thenkey~} #2}%
```

`\setcoplienkeys` \setportlandkeys{⟨*problem*⟩}{⟨*context*⟩}{⟨*forces*⟩}{⟨*solution*⟩}{⟨*reasoning*⟩}{⟨*resultingcontext*⟩} sets the values for if-then when writing a Portland-style pattern

```
\NewDocumentCommand{\setcoplienkeys}{mmmmmm}{%
    \def\coplien@key@prob{#1}%
    \def\coplien@key@cont{#2}%
    \def\coplien@key@forc{#3}%
    \def\coplien@key@solu{#4}%
    \def\coplien@key@reas{#5}%
    \def\coplien@key@cnrs{#6}%
}%
\setcoplienkeys{\textbf{Problema:}}{\textbf{Contexto:}}{\textbf{Forças:}}{\textbf{Solução:}}{\t
```

`\coplien` \coplien {⟨*problem*⟩}{⟨*context*⟩}{⟨*forces*⟩}{⟨*solution*⟩}{⟨*reasoning*⟩}{⟨*resultingcontext*⟩}

Writes a pattern in Coplien format

```
\NewDocumentCommand{\coplien}{mmmmmm}{%
    \textbf{\coplien@key@prob}: #1 \par%
    \textbf{\coplien@key@cont}: #2\par%
    \textbf{\coplien@key@forc}: #3 \par%
    \textbf{\coplien@key@solu}: #4 \par%
    \textbf{\coplien@key@reas}: #5 \par%
    \textbf{\coplien@key@cnrs}: #6 \par%
}%
```

5.7 Index making

```
\if@showindex%
\makeindex%
\fi%
```

References

[Wik] C2 Wiki. *Pattern Definition Thread*. URL: `https://wiki.c2.com/?PatternDefinitionThread` (visited on 10/22/2021).