The eventB package*

Thai Son Hoang
ETH-Zurich
<htson at inf dot ethz dot ch>

February 26, 2013

Abstract

This class provides a template for typesetting Event-B models. It was developed at the Swiss Federal Institute of Technology Zurich (ETH-Zurich).

Contents

1 Introduction **1**

2 Usage **1**
2.1 Package Options 1

3 Implementation **2**
3.1 Package Loading 2
3.2 Declaration of Options for the Package 2
3.3 Commands for Pretty-Print Event-B Models 4

1 Introduction

This package was developed in order to ease the typesetting of Event-B models in LaTeX.

2 Usage

See `sample-eventB.tex` for an example of how to use the package.

2.1 Package Options

The package offers the following options:

- `nobox`: to disable to bounding boxes for the Event-B modelling elements,
- `small`, `compact`, `tiny`: options for font size,
- `colour` (or `color`): to colour several modelling elements.

*This document corresponds to eventB v1.1.1, dated 2012/02/21.

3 Implementation

3.1 Package Loading

We begin by loading the required package `xspace` and `xcolor`.

```
\RequirePackage{xspace}
\RequirePackage{xcolor}
\RequirePackage{ifthen}
\RequirePackage{etoolbox}
```

3.2 Declaration of Options for the Package

In this part various options for the package are defined.

Option for rounding boxes By default, Event-B modelling elements, e.g., invariants, events, etc., are displayed in a rounding box. This option enables them to be displayed without the rounding box.

```
\newlength{\B@oldfboxsep}
\newcommand{\event}[7][]{
  \setlength{\B@oldfboxsep}{\fboxsep}
  \setlength{\fboxsep}{2ex}
  \fbox{
    \ensuremath{
      \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
    }
  }
  \setlength{\fboxsep}{\B@oldfboxsep}
}

\newcommand{\B@declaration}[2]{
  \setlength{\B@oldfboxsep}{\fboxsep}
  \setlength{\fboxsep}{2ex}
  \fbox{
    \ensuremath{
      \B@declarationbase{#1}{#2}
    }
  }
  \setlength{\fboxsep}{\B@oldfboxsep}
}

\newcommand{\B@section}[3][]{
  \setlength{\B@oldfboxsep}{\fboxsep}
  \setlength{\fboxsep}{2ex}
  \fbox{
    \ensuremath{
      \B@sectionbase[#1]{#2}{#3}
    }
  }
  \setlength{\fboxsep}{\B@oldfboxsep}
}

\DeclareOption{nobox}{
  \renewcommand{\event}[7][]{
```

```
    \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
  }

  \renewcommand{\B@declaration}[2]{
    \B@declarationbase{#1}{#2}
  }

  \renewcommand{\B@section}[3][]{
    \B@sectionbase[#1]{#2}{#3}
  }
}

```

Options for font size and spacing We define the default values for font size and some spacing commands, and how the are redefined according to options `small`, `compact`, and `tiny`. In particular, option `compact` and `tiny` implies option `nobox`.

```
\newcommand{\B@fontsize}{\normalsize} % The font size used in Bcode environment
\newcommand{\Bvspace}[1][2ex]{\\[#1]} % Vertical space
\newcommand{\Bhspace}[1][2em]{\hspace{#1}} % Horizontal space
\newcommand{\B@tab}{\quad} % A small separation space

\DeclareOption{small}{
  \renewcommand{\B@fontsize}{\small}
  \renewcommand{\Bvspace}[1][1ex]{\\[#1]}
  \renewcommand{\Bhspace}[1][1em]{\hspace{#1}}
  \renewcommand{\B@tab}{\ }
}
\DeclareOption{compact}{
  \renewcommand{\B@fontsize}{\footnotesize}
  \renewcommand{\Bvspace}[1][0ex]{\\[#1]}
  \renewcommand{\Bhspace}[1][0.5em]{\hspace{#1}}
  \renewcommand{\B@tab}{\ }
  \ExecuteOptions{nobox}
}
\DeclareOption{tiny}{
  \renewcommand{\B@fontsize}{\scriptsize}
  \renewcommand{\Bvspace}[1][-0.5ex]{\\[#1]}
  \renewcommand{\Bhspace}[1][0.5em]{\hspace{#1}}
  \renewcommand{\B@tab}{\ }
  \ExecuteOptions{nobox}
}

```

Options for colouring Keywords, labels and identifiers in Event-B can be coloured. We define several commands and redefine them accordingly for colouring. When `colour` (or `color`) option is enabled, one can customise the colours for Event-B keywords, labels or identifier or proof obligation labels.

```
\newcommand{\B@keyword}[1]{\ensuremath{\B@keywordbase{#1}}\xspace}
\newcommand{\Bidentifier}[1]{\ensuremath{\B@identifier{#1}}\xspace}
\newcommand{\Blabel}[2][]{\ensuremath{\B@label[#1]{#2}}\xspace}
\newcommand{\Bpo}[1]{\ensuremath{\B@po{#1}}\xspace}
```

```
\DeclareOption{colour}{
  \newcommand{\setBKeywordColour}[1]{\colorlet{B@keywordcolor}{#1}}
  \setBKeywordColour{blue}
  \newcommand{\setBIdentifierColour}[1]{\colorlet{B@identifiercolor}{#1}}
  \setBIdentifierColour{blue!50!red}
  \newcommand{\setBLabelColour}[1]{\colorlet{B@labelcolor}{#1}}
  \setBLabelColour{green!50!black}
  \newcommand{\setBPOColour}[1]{\colorlet{B@pocolor}{#1}}
  \setBPOColour{red}
  \renewcommand{\B@keyword}[1]{
    \ensuremath{\textcolor{B@keywordcolor}{\B@keywordbase{#1}}}\xspace
  }
  \renewcommand{\Bidentifier}[1]{
    \ensuremath{\textcolor{B@identifiercolor}{\B@identifier{#1}}}\xspace
  }
  \renewcommand{\Blabel}[2][]{
    \ensuremath{\textcolor{B@labelcolor}{\B@label[#1]{#2}}}\xspace
  }
  \renewcommand{\Bpo}[1]{
    \ensuremath{\textcolor{B@pocolor}{\B@po{#1}}}\xspace
  }
}
\DeclareOption{color}{
  \ExecuteOptions{colour}
}

```

After declaration of options, we execute them accordingly.

```
\ProcessOptions
```

3.3 Commands for Pretty-Print Event-B Models

We start with the definition of the `\eventB` macro.

```
\newcommand{\eventB}{Event-B\xspace}
```

The `Bcode` environment for displaying Event-B models. The environment has an optional argument for specifying the font size. By default, it is the same as the `\B@fontsize` controlled by the package option.

```
\newenvironment{Bcode}[1][\B@fontsize]{\begin{center}#1}{\end{center}}
```

Declarations and Collections Event-B modelling elements are organised into declarations (e.g., variables, constants, etc.) or collections (e.g., invariants, axioms). For each declaration, the input is a comma-separated list of elements. For each collection, the input is a newly(`\\`)-separated list of elements.

```
\newcommand{\carriersets}[1]{
  \B@declaration{sets}{#1}
}
\newcommand{\constants}[1]{
  \B@declaration{constants}{#1}
}
\newcommand{\axioms}[2][]{
  \B@section[#1]{axioms}{#2}
}
```

```
\newcommand{\variables}[1]{
  \B@declaration{variables}{#1}
}
\newcommand{\invariants}[2][]{
  \B@section[#1]{invariants}{#2}
}
\newcommand{\variant}[1]{
  \B@declaration{variant}{#1}
}
```

Event-B keywords We define the keywords for pretty-print Event-B models.

```
\newcommand{\Bany}{\B@keyword{any}}
\newcommand{\Bbegin}{\B@keyword{begin}}
\newcommand{\Bend}{\B@keyword{end}}
\newcommand{\Brefines}{\B@keyword{refines}}
\newcommand{\Bstatus}{\B@keyword{status}}
\newcommand{\Bthen}{\B@keyword{then}}
\newcommand{\Bwhen}{\B@keyword{when}}
\newcommand{\Bwhere}{\B@keyword{where}}
\newcommand{\Bwith}{\B@keyword{with}}
```

Event-B modelling elements We define several macros for pretty-print Event-B modelling elements.

```
\newcommand{\Bctx}[1]{\ensuremath{\mathbf{#1}}\xspace}
\newcommand{\Bset}[1]{\Bidentifier{#1}}
\newcommand{\Bcst}[1]{\Bidentifier{#1}}
\newcommand{\Baxm}[1]{\Blabel{#1}}
\newcommand{\Bthm}[1]{\Blabel[thm]{#1}}

\newcommand{\Bmch}[1]{\ensuremath{\mathbf{#1}}\xspace}
\newcommand{\Bvrb}[1]{\Bidentifier{#1}}
\newcommand{\Binv}[1]{\Blabel{#1}}
\newcommand{\Bevt}[1]{\Blabel{#1}}
\newcommand{\Bpar}[1]{\Bidentifier{#1}}
\newcommand{\Bact}[1]{\Blabel{#1}}
\newcommand{\Bgrd}[1]{\Blabel{#1}}
\newcommand{\Bbap}[1]{\hbox{\sl\bfseries #1}}
```

Meta-macros for creating macros for modelling elements We define meta-marcos to create macros for different modelling elements.

```
\newcommand{\B@newmacro}[3][]{
  \ifstrequal{#1}{}{
    \expandafter\def\csname #2\endcsname{#3{#2}}
  }{
    \expandafter\def\csname #1\endcsname{#3{#2}}
  }
}
\newcommand{\newBctx}[2][]{\B@newmacro[#1]{#2}{\Bctx}}
\newcommand{\newBset}[2][]{\B@newmacro[#1]{#2}{\Bset}}
\newcommand{\newBcst}[2][]{\B@newmacro[#1]{#2}{\Bcst}}
```

```
\newcommand{\newBaxm}[2][]{\B@newmacro[#1]{#2}{\Baxm}}
\newcommand{\newBthm}[2][]{\B@newmacro[#1]{#2}{\Bthm}}
\newcommand{\newBmch}[2][]{\B@newmacro[#1]{#2}{\Bmch}}
\newcommand{\newBvrb}[2][]{\B@newmacro[#1]{#2}{\Bvrb}}
\newcommand{\newBinv}[2][]{\B@newmacro[#1]{#2}{\Binv}}
\newcommand{\newBevt}[2][]{\B@newmacro[#1]{#2}{\Bevt}}
\newcommand{\newBpar}[2][]{\B@newmacro[#1]{#2}{\Bpar}}
\newcommand{\newBgrd}[2][]{\B@newmacro[#1]{#2}{\Bgrd}}
\newcommand{\newBact}[2][]{\B@newmacro[#1]{#2}{\Bact}}

%%%% Theorem Proof Obligation
%%%% Print the theorem proof obligation, given the theorem label.
%%%% Arguments:
%%%% 1. Theorem label
%%%%
%%%% Usage:
%%%% - \thmpo{thm} will produce "thm/THM"
\newcommand{\thmpo}[1]{\Bthm{#1}/\Bpo{THM}}

%%%% Axiom Well-definedness Proof Obligation
%%%% Print the axiom well-definedness proof obligation, given the
%%%% axiom label.
%%%% Arguments:
%%%% 1. Axiom label
%%%%
%%%% Usage:
%%%% - \axmwdpo{axm} will produce "axm/WD"
\newcommand{\axmwdpo}[1]{\Baxm{#1}/\Bpo{WD}}

%%%% Invariant Proof Obligation
%%%% Print the invariant proof obligation, given the event name and
%%%% invariant label
%%%% Arguments:
%%%% 1. Event name
%%%% 2. Invariant label
%%%%
%%%% Usage:
%%%% - \invpo{evt}{inv} will produce "evt/inv/INV"
\newcommand{\invpo}[2]{\Bevt{#1}/\Binv{#2}/\Bpo{INV}}

%%%% Theorem (in guard) Proof Obligation
%%%% Print the simulation proof obligation, given the event name and
%%%% the theorem (in guard) label.
%%%% Arguments:
%%%% 1. Event name
%%%% 2. Theorem (in guard) label
%%%%
%%%% Usage:
%%%% - \grdthmpo{evt}{thm} will produce "evt/thm/THM"
```

```
\newcommand{\grdthmpo}[2]{\Bevt{#1}/\Bthm{#2}/\Bpo{THM}}

%%%% Feasibility Proof Obligation
%%%% Print the feasibility proof obligation, given the event name and
%%%% the action label
%%%% Arguments:
%%%% 1. Event name
%%%% 2. Action label
%%%%
%%%% Usage:
%%%% - \fispo{evt}{act} will produce "evt/act/FIS"
\newcommand{\fispo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{FIS}}

%%%% Variant finiteness Proof Obligation
%%%% Print the Variant finiteness proof obligation
%%%% Arguments: No arguments
%%%%
%%%% Usage:
%%%% - \finpo will produce "FIN"
\newcommand{\finpo}{\Bpo{FIN}}

%%%% Variant Proof Obligation
%%%% Print the guard strengthen proof obligation, given the event name
%%%% Arguments:
%%%% 1. Event name
%%%%
%%%% Usage:
%%%% - \grdpo{evt} will produce "evt/VAR"
\newcommand{\varpo}[1]{\Bevt{#1}/\Bpo{VAR}}

%%%% Simulation Proof Obligation
%%%% Print the simulation proof obligation, given the event name and
%%%% the action label.
%%%% Arguments:
%%%% 1. Event name
%%%% 2. Action label
%%%%
%%%% Usage:
%%%% - \simpo{evt}{act} will produce "evt/act/SIM"
\newcommand{\simpo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{SIM}}

%%%% Guard Strengthen Proof Obligation
%%%% Print the guard strengthen proof obligation, given the event
%%%% name and the guard label
%%%% Arguments:
%%%% 1. (Abstract) Event name
%%%% 2. (Abstract) Guard label
%%%%
%%%% Usage:
%%%% - \grdpo{evt}{grd} will produce "evt/grd/GRD"
\newcommand{\grdpo}[2]{\Bevt{#1}/\Bgrd{#2}/\Bpo{GRD}}
```

```
%%%%% Variant Natural Number Proof Obligation
%%%%% Print the Variant Natural Number proof obligation, given the event name
%%%%% Arguments:
%%%%% 1. Event name
%%%%%
%%%%% Usage:
%%%%% - \natpo{evt} will produce "evt/NAT"
\newcommand{\natpo}[1]{\Bevt{#1}/\Bpo{NAT}}

\newcommand{\B@keywordbase}[1]{\mathbf{#1}}
\newcommand{\B@identifier}[1]{\mathit{#1}}
\newcommand{\B@label}[2][]{
  \ifstrequal{#1}{}{
    \mathsf{#2}
  }{
    \mathit{#2}
  }
}



\newcommand{\eventinline}[7][]{
  \setlength{\B@oldfboxsep}{\fboxsep}
  \setlength{\fboxsep}{2ex}
  \fbox{
    \ensuremath{
      \B@eventinline[#1]{#2}{#3}{#4}{#5}{#6}{#7}
    }
  }
  \setlength{\fboxsep}{\B@oldfboxsep}
}


\newcommand{\B@declarationbase}[2]{
  \begin{array}{l@{\B@tab}l}
    \B@keyword{#1:} &  #2
  \end{array}
}

\newcommand{\B@sectionbase}[3][]{
  \ifstrequal{#1}{}{
    \begin{array}{l}
      \B@keyword{#2:} \\
      \begin{array}{l@{\B@tab}l}
        #3
      \end{array}
    \end{array}
  }{
    \begin{array}{l@{\B@tab}l}
      #3
    \end{array}
  }
}

```

```
\newcommand{\B@po}[1]{\ensuremath{\mathsf{#1}}\xspace}

%%%% (BEGIN) Macros for Pretty-Print Event-B Components %%%
\newcommand{\SKIP}{\textsc{skip}}


%%%% Pretty print an general Event-B event
%%%% Arguments:
%%%% 1. (Optional) convergence status.
%%%% 2. Name of the event.
%%%% 3. Name of the abstract event.
%%%% 4. (Comma-separated) list of parameters.
%%%% 5. (Newline(\\)-separated) list of guards.
%%%% 6. (Newline(\\)-separated) list of witness predicates.
%%%% 7. (Newline(\\)-separated) list of assignments.
%%%%
%%%% Usage: \B@event[conv]{conc}{abs}{x,y}{G1(x,y)\\G2(x,y)}{W1\\W2}{S1(v,x,y)\\S2(w,x,y)}
%%%%       will produce the following
%%%%
%%%%       conc
%%%%       refines abs
%%%%       status conv
%%%%       any x, y where
%%%%         G1(x, y)
%%%%         G2(x, y)
%%%%       with
%%%%         W1
%%%%         W2
%%%%       then
%%%%         S1(v, x, y)
%%%%         S2(w, x, y)
%%%%       end
%%%%
%%%% Special case:
%%%% - Empty abstract event --> refines clause is omitted.
%%%% - Empty convergence status --> status clause is omitted.
%%%% - Empty witness --> with clause is omitted.
%%%% - Empty parameters, empty guards --> begin ... end
%%%% - Empty parameters --> when ... then ... end
%%%% - Empty actions --> \SKIP
\newcommand{\B@ifstrequal}{\expandafter\ifstrequal\expandafter}
\newcommand{\B@event}[7][]{%
  {
  \newcommand\evt@sts{#1}%
  \newcommand\evt@name{#2}%
  \newcommand\evt@absevts{#3}%
  \newcommand\evt@pars{#4}%
  \newcommand\evt@grds{#5}%
  \newcommand\evt@wits{#6}
  \newcommand\evt@acts{#7}
  %% Pretty-print convergence status
  \B@ifstrequal{\evt@sts}{}{
    \newcommand\pretty@sts{}
```

```
}{
  \newcommand\pretty@sts{\B@tab\Bstatus \B@tab \evt@sts \\}
}
% Pretty-print abstract events
\B@ifstrequal{\evt@absevts}{}{
  \newcommand\pretty@absevts{}
}{
  \newcommand\pretty@absevts{\B@tab\Brefines \B@tab \evt@absevts{} \\}%
}
% Pretty-print parameters
\B@ifstrequal{\evt@pars}{}{
  \newcommand\pretty@pars{}
}{
  \newcommand\pretty@pars{\B@tab\Bany \B@tab \evt@pars \B@tab \Bwhere \\}
}
% Pretty-print guards
\B@ifstrequal{\evt@grds}{}{
  \newcommand\pretty@grds{}
}{
  \newcommand\pretty@grds@tmp{
    \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
      \evt@grds
    \end{array}\\
  }
  \B@ifstrequal{\evt@pars}{}{
    \newcommand\pretty@grds{
      \B@tab \Bwhen \\
      \pretty@grds@tmp
    }
  }{
    \newcommand\pretty@grds{\pretty@grds@tmp}
  }
}
% Pretty-print witnesses
\B@ifstrequal{\evt@wits}{}{
  \newcommand\pretty@wits{}
}{
  \newcommand\pretty@wits{
    \B@tab\Bwith\\
    \begin{array}{@{\B@tab\B@tab}ll}
      \evt@wits
    \end{array}\\
  }
}
% Pretty-print actions
\B@ifstrequal{\evt@acts}{}{
  \renewcommand\evt@acts{\SKIP}
}{}
\newcommand\pretty@acts@tmp{
  \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
    \evt@acts
  \end{array}\\
}
\newcommand\pretty@acts@keyword{\B@tab\Bthen \\}
```

```
  \B@ifstrequal{\evt@pars}{}{
    \B@ifstrequal{\evt@grds}{}{
      \renewcommand\pretty@acts@keyword{\B@tab\Bbegin \\}
    }{}
  }{}
  \newcommand\pretty@acts{
    \pretty@acts@keyword
    \pretty@acts@tmp
  }
  % Really do it now
  \begin{array}{l}
    \Bevt{\evt@name} \\
    \pretty@sts
    \pretty@absevts
    \pretty@pars
    \pretty@grds
    \pretty@wits
    \pretty@acts
    \B@tab\Bend
  \end{array}
}
}

%%%% Pretty print an general Event-B event
%%%% Arguments:
%%%% 1. (Optional) convergence status.
%%%% 2. Name of the event.
%%%% 3. Name of the abstract event.
%%%% 4. (Comma-separated) list of parameters.
%%%% 5. (Newline(\\)-separated) list of guards.
%%%% 6. (Newline(\\)-separated) list of witness predicates.
%%%% 7. (Newline(\\)-separated) list of assignments.
%%%%
%%%% Usage: \B@event[conv]{conc}{abs}{x,y}{G1(x,y)\\G2(x,y)}{W1\\W2}{S1(v,x,y)\\S2(w,x,y)}
%%%%       will produce the following
%%%%
%%%%       conc
%%%%       refines abs
%%%%       status conv
%%%%       any x, y where
%%%%         G1(x, y)
%%%%         G2(x, y)
%%%%       with
%%%%         W1
%%%%         W2
%%%%       then
%%%%         S1(v, x, y)
%%%%         S2(w, x, y)
%%%%       end
%%%%
%%%% Special case:
%%%% - Empty abstract event --> refines clause is omitted.
%%%% - Empty convergence status --> status clause is omitted.
%%%% - Empty witness --> with clause is omitted.
```

```
%%%%% - Empty parameters, empty guards --> begin ... end
%%%%% - Empty parameters --> when ... then ... end
%%%%% - Empty actions --> \SKIP
\newcommand{\B@eventinline}[7][]{
  \def\evt@sts{#1}
  \def\evt@name{#2}
  \def\evt@absevts{#3}
  \def\evt@pars{#4}
  \def\evt@grds{#5}
  \def\evt@wits{#6}
  \def\evt@acts{#7}
  %% Ignore convergence status
  \def\pretty@sts{}
  % Pretty-print abstract events
  \ifx\evt@absevts\@empty
  \def\pretty@absevts{}
  \else
  \def\pretty@absevts{\Brefines~\evt@absevts~}
  \fi
  % Pretty-print parameters
  \ifx\evt@pars\@empty
  \def\pretty@pars{}
  \else
  \def\pretty@pars{\Bany~\evt@pars~\Bwhere~}
  \fi
  % Pretty-print guards
  \ifx\evt@grds\@empty
  \def\pretty@grds{}
  \else
  \def\evt@grds@tmp{
      \evt@grds
  }
  \ifx\evt@pars\@empty
  \def\pretty@grds{
    \Bwhen~
    \evt@grds@tmp~
  }
  \else
  \def\pretty@grds{\evt@grds@tmp~}
  \fi
  \fi
  % Pretty-print witnesses
  \ifx\evt@wits\@empty
  \def\pretty@wits{}
  \else
  \def\pretty@wits{
    \Bwith~
    \evt@wits~
  }
  \fi
  % Pretty-print actions
  \ifx\evt@acts\@empty
  \def\evt@acts{\SKIP}
  \else
```

```
  \fi
  \def\evt@acts@tmp{
    \evt@acts
  }
  \def\evt@acts@keyword{\Bthen}
  \ifx\evt@pars\@empty
  \ifx\evt@grds\@empty
  \def\evt@acts@keyword{\Bbegin}
  \else
  \fi
  \else
  \fi
  \def\pretty@acts{
    \evt@acts@keyword~
    \evt@acts@tmp~
  }
  % Really do it now
  \begin{array}{l}
    \Bevt{\evt@name}~\widehat{=}~
    \pretty@sts
    \pretty@absevts
    \pretty@pars
    \pretty@grds
    \pretty@wits
    \pretty@acts
    \Bend
  \end{array}
}

%%%% INITIALISATION label
\newBevt{init}

%%%% Pretty print the initialisation: no ``refines'' clause. no parameters, no
%%%% guards
%%%% Arguments:
%%%% 1. (Newline(\\)-separated) list of assignments.
%%%%
%%%% Usage: \init{S1(v,x,y)\\S2(w,x,y)}
%%%%       will produce the following
%%%%
%%%%       init
%%%%       begin
%%%%         S1(v, x, y)
%%%%         S2(w, x, y)
%%%%       end
%%%%
\newcommand{\initialisation}[1]{
  \event{\init}{}{}{}{}{#1}
}
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\@empty 493, 499, 505, 511, 521, 530, 538, 539
\\ 54, 60, 66, 73, 306, 330–332, 334, 372, 378, 384, 393, 397, 409, 412, 422, 424, 427, 436, 454–456, 458, 568, 570
\␣ 62, 68, 75

A

\axioms 118
\axmwdpo 189, 190

B

\B@declaration 17, 44, 113, 116, 122, 128
\B@declarationbase 22, 45, 297
\B@event 11, 41, 334, 359, 458
\B@eventinline 290, 482
\B@fontsize 53, 59, 65, 72, 111
\B@identifier 80, 96, 274
\B@ifstrequal 358, 369, 375, 381, 387, 395, 405, 416, 425, 426
\B@keyword . . 79, 92, 130–138, 299, 306
\B@keywordbase 79, 93, 273
\B@label . . . 81, 99, 275
\B@newmacro 153, 160–171
\B@oldfboxsep 5, 7, 14, 18, 25, 29, 36, 286, 293
\B@po 82, 102, 318
\B@section 28, 48, 119, 125
\B@sectionbase 33, 49, 303
\B@tab 56, 62, 68, 75, 298, 307, 312, 372, 378, 384, 391, 397, 409, 410, 420, 424, 427, 443
\Bact . 150, 171, 223, 251
\Bany 130, 384, 502
\Baxm 142, 163, 190
\Bbap 152
\Bbegin . . . 131, 427, 540
\Bcst 141, 162
\Bctx 139, 160
\begin 111, 298, 305, 307, 312, 391, 410, 420, 435, 550
\Bend 132, 443, 558
\Bevt 148, 168, 201, 212, 223, 240, 251, 262, 271, 436, 551
\bfseries 152
\Bgrd 151, 170, 262
\Bhspace . 55, 61, 67, 74
\Bidentifier . 80, 95, 140, 141, 146, 149
\Binv 147, 167, 201
\Blabel 81, 98, 142, 143, 147, 148, 150, 151
\Bmch 145, 165
\Bpar 149, 169
\Bpo 82, 101, 180, 190, 201, 212, 223, 231, 240, 251, 262, 271
\Brefines . 133, 378, 496
\Bset 140, 161
\Bstatus 134, 372
\Bthen 135, 424, 537
\Bthm . 143, 164, 180, 212
\Bvrb 146, 166
\Bvspace . 54, 60, 66, 73
\Bwhen 136, 397, 513
\Bwhere . . . 137, 384, 502
\Bwith 138, 409, 525

C

\carriersets 112
\colorlet 84, 86, 88, 90
\constants 115
\csname 155, 157

D

\DeclareOption . 39, 58, 64, 71, 83, 105
\def 155, 157, 483–489, 491, 494, 496, 500, 502, 506, 508, 512, 517, 522, 524, 531, 534, 537, 540, 545

E

\else 495, 501, 507, 516, 523, 532, 541, 543
\end . . 111, 300, 309, 310, 314, 393, 412, 422, 444, 559
\endcsname . . . 155, 157
\ensuremath 10, 21, 32, 79–82, 93, 96, 99, 102, 139, 145, 289, 318
\event 6, 40, 580
\eventB 110
\eventinline 285
\evt@absevts 363, 375, 378, 485, 493, 496
\evt@acts 367, 416, 417, 421, 489, 530, 531, 535
\evt@acts@keyword 537, 540, 546
\evt@acts@tmp . 534, 547
\evt@grds 365, 387, 392, 426, 487, 505, 509, 539
\evt@grds@tmp 508, 514, 517
\evt@name 362, 436, 484, 551
\evt@pars 364, 381, 384,

395, 425, 486, 499, 502, 511, 538
\evt@sts 361, 369, 372, 483
\evt@wits . 366, 405, 411, 488, 521, 526
\ExecuteOptions 69, 76, 106
\expandafter 155, 157, 358

F
\fbox 9, 20, 31, 288
\fboxsep . . . 7, 8, 14, 18, 19, 25, 29, 30, 36, 286, 287, 293
\fi 497, 503, 518, 519, 528, 533, 542, 544
\finpo 230, 231
\fispo 222, 223
\footnotesize 65

G
\grdpo 239, 261, 262
\grdthmpo 211, 212

H
\hbox 152
\hspace . . 55, 61, 67, 74

I
\ifstrequal 154, 276, 304, 358
\ifx 493, 499, 505, 511, 521, 530, 538, 539
\init 570, 580
\initialisation . . . 579
\invariants 124
\invpo 200, 201

M
\mathbf . . . 139, 145, 273
\mathit 274, 279
\mathsf 277, 318

N
\natpo 270, 271
\newBact 171
\newBaxm 163
\newBcst 162
\newBctx 160
\newBevt 168, 563
\newBgrd 170
\newBinv 167
\newBmch 165
\newBpar 169
\newBset 161
\newBthm 164
\newBvrb 166
\newcommand 6, 17, 28, 53–56, 79–82, 84, 86, 88, 90, 110, 112, 115, 118, 121, 124, 127, 130–143, 145–153, 160–171, 180, 190, 201, 212, 223, 231, 240, 251, 262, 271, 273–275, 285, 297, 303, 318, 321, 358, 359, 361–367, 370, 372, 376, 378, 382, 384, 388, 390, 396, 401, 406, 408, 419, 424, 430, 482, 579
\newenvironment . . . 111
\newlength 5
\normalsize 53

P
\pretty@absevts 376, 378, 438, 494, 496, 553
\pretty@acts 430, 442, 545, 557
\pretty@acts@keyword 424, 427, 431
\pretty@acts@tmp 419, 432
\pretty@grds . . 388, 396, 401, 440, 506, 512, 517, 555
\pretty@grds@tmp 390, 398, 401
\pretty@pars 382, 384, 439, 500, 502, 554
\pretty@sts . . . 370, 372, 437, 491, 552
\pretty@wits 406, 408, 441, 522, 524, 556
\ProcessOptions . . . 109

Q
\quad 56

R
\renewcommand 40, 44, 48, 59–62, 65–68, 72–75, 92, 95, 98, 101, 417, 427
\RequirePackage . . . 1–4

S
\scriptsize 72
\setBIdentifierColour 86, 87
\setBKeywordColour 84, 85
\setBLabelColour 88, 89
\setBPOColour . . . 90, 91
\setlength . 7, 8, 14, 18, 19, 25, 29, 30, 36, 286, 287, 293
\simpo 250, 251
\SKIP 321, 357, 417, 481, 531
\sl 152
\small 59

T
\textcolor 93, 96, 99, 102
\textsc 321
\thmpo 179, 180

V
\variables 121
\variant 127
\varpo 240

W
\widehat 551

X
\xspace 79–82, 93, 96, 99, 102, 110, 139, 145, 318

Change History

v1.0
General: Initial version 1
v1.0.1
General: Ensure that the keywords, labels are in math-mode 1
v1.1
General: Re-implement how options are defined, added options 'box' 1
v1.1.1
General: Updated documentation . 1