

The `eventB` package*

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

February 27, 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	2
3	Implementation	2
3.1	Package Loading	2
3.2	Helper Macros	2
3.3	Declaration of Options for the Package	6
3.3.1	Option for bounding boxes	6
3.3.2	Options for font size and spacing	7
3.3.3	Options for colouring	9
3.4	Commands for Pretty-Print Event-B Models	9

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.

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

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.

3 Implementation

3.1 Package Loading

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

```
1 \RequirePackage{xspace}
2 \RequirePackage{xcolor}
3 \RequirePackage{etoolbox}
```

3.2 Helper Macros

We define same basic helper macros that will be used to defined other macros.

```
\B@keywordbase Basic macro for Event-B keywords.
4 \newcommand{\B@keywordbase}[1]{\mathbf{#1}}

\B@identifierbase Basic macro for Event-B identifiers.
5 \newcommand{\B@identifierbase}[1]{\mathit{#1}}

\B@declarationbase Basic macro for Event-B declarations (e.g., variables, constants, etc.).
6 \newcommand{\B@declarationbase}[2]{
7   \begin{array}{l@{\B@tab}l}
8     \B@keyword{#1:} & #2
9   \end{array}
10 }

\B@sectionbase Basic macro for Event-B sections (e.g., invariants, axioms, etc.).
11 \newcommand{\B@sectionbase}[3][ ]{
12   \ifstrequal{#1}{ }{
13     \begin{array}{l}
14       \B@keyword{#2:} \\\
15       \begin{array}{l@{\B@tab}l}
16         #3
17       \end{array}
18     \end{array}
19   }{
20     \begin{array}{l@{\B@tab}l}
21       #3
22     \end{array}
23   }
24 }

\B@ifstrequal A wrapper for ake sure that the first argument is properly expanded.
25 \newcommand{\B@ifstrequal}{\expandafter\ifstrequal\expandafter}
```

```

\B@eventbase Basic macro for pretty-print Event-B events.
26 \newcommand{\B@eventbase}[7] [] {%
27   { % BEGIN group
We first save the arguments to local variables.
28   \newcommand\evt@sts{#1}% Event status
29   \newcommand\evt@label{#2}% Event label
30   \newcommand\evt@absevt{#3}% Abstract event
31   \newcommand\evt@pars{#4}% Event parameters
32   \newcommand\evt@grds{#5}% Event guards
33   \newcommand\evt@wits{#6}% Event witnesses
34   \newcommand\evt@acts{#7}% Event actions
The convergence status is skipped if empty.
35   \B@ifstrequal{\evt@sts}{}{
36     \newcommand\pretty@sts{}
37   }{
38     \newcommand\pretty@sts{\B@tab\Bstatus \B@tab \evt@sts \\\}
39   }
The refines clause is skipped if there are no abstract events.
40   \B@ifstrequal{\evt@absevt}{}{
41     \newcommand\pretty@absevt{}
42   }{
43     \newcommand\pretty@absevt{\B@tab\Brefines \B@tab \evt@absevt{} \\\}
44   }
The parameters is skipped if there are none.
45   \B@ifstrequal{\evt@pars}{}{
46     \newcommand\pretty@pars{}
47   }{
48     \newcommand\pretty@pars{\B@tab\Bany \B@tab \evt@pars \B@tab \Bwhere \\\}
49   }
The keywords for guards also depends on if there are parameters or not.
50   \B@ifstrequal{\evt@grds}{}{
51     \newcommand\pretty@grds{}
52   }{
53     \newcommand\pretty@grds@tmp{
54       \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
55       \evt@grds
56       \end{array}\\\}
57   }
58   \B@ifstrequal{\evt@pars}{}{
59     \newcommand\pretty@grds{
60       \B@tab \Bwhen \\\}
61     \pretty@grds@tmp
62   }
63   }{
64     \newcommand\pretty@grds{\pretty@grds@tmp}
65   }
66   }
The witnesses are skipped if there are none.
67   \B@ifstrequal{\evt@wits}{}{
68     \newcommand\pretty@wits{}

```

```

69   }{
70     \newcommand\pretty@wits{
71       \B@tab\Bwith\\
72       \begin{array}{@{\B@tab\B@tab}ll}
73         \evt@wits
74       \end{array}\\
75     }
76   }

```

When there are no actions, SKIP is used. The keyword is changed depending on whether the event has parameters or not.

```

77   \B@ifstrequal{\evt@acts}{}{
78     \renewcommand\evt@acts{\SKIP}
79   }{}
80   \newcommand\pretty@acts@tmp{
81     \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
82       \evt@acts
83     \end{array}\\
84   }
85   \newcommand\pretty@acts@keyword{\B@tab\Bthen \\}
86   \B@ifstrequal{\evt@pars}{}{
87     \B@ifstrequal{\evt@grds}{}{
88       \renewcommand\pretty@acts@keyword{\B@tab\Bbegin \\}
89     }{}
90   }{}
91   \newcommand\pretty@acts{
92     \pretty@acts@keyword
93     \pretty@acts@tmp
94   }

```

Finally we put all the pretty-print pieces together.

```

95   \begin{array}{l}
96     \Bevt{\evt@label} \\
97     \pretty@sts
98     \pretty@absevt
99     \pretty@pars
100    \pretty@grds
101    \pretty@wits
102    \pretty@acts
103    \B@tab\Bend
104  \end{array}
105 } % END group
106 }

```

\B@inlineeventbase Basic macro for pretty-print Event-B events inline.

```

107 \newcommand{\B@inlineeventbase}[7][]{
108   { % BEGIN group

```

We first save the arguments to local variables.

```

109     \newcommand\evt@sts{#1}% Event status
110     \newcommand\evt@label{#2}% Event label
111     \newcommand\evt@absevt{#3}% Abstract event
112     \newcommand\evt@pars{#4}% Event parameters
113     \newcommand\evt@grds{#5}% Event guards
114     \newcommand\evt@wits{#6}% Event witnesses
115     \newcommand\evt@acts{#7}% Event actions

```

The convergence status is skipped if empty.

```

116   \B@ifstrequal{\evt@sts}{-}{-}{
117     \newcommand\pretty@sts{}
118   }{
119     \newcommand\pretty@sts{(\evt@sts)}
120   }

```

The refines clause is skipped if there are no abstract events.

```

121   \B@ifstrequal{\evt@absevt}{-}{-}{
122     \newcommand\pretty@absevt{}
123   }{
124     \newcommand\pretty@absevt{\Brefines~\evt@absevt}
125   }

```

The parameters is skipped if there are none.

```

126   \B@ifstrequal{\evt@pars}{-}{-}{
127     \newcommand\pretty@pars{}
128   }{
129     \newcommand\pretty@pars{\Bany~\evt@pars~\Bwhere~}
130   }

```

The keywords for guards also depends on if there are parameters or not.

```

131   \B@ifstrequal{\evt@grds}{-}{-}{
132     \newcommand\pretty@grds{}
133   }{
134     \newcommand\pretty@grds@tmp{
135       \evt@grds~
136     }
137     \B@ifstrequal{\evt@pars}{-}{-}{
138       \Bwhen~\pretty@grds@tmp
139     }{
140       \newcommand\pretty@grds{\pretty@grds@tmp}
141     }
142   }

```

The witnesses are skipped if there are none.

```

143   \B@ifstrequal{\evt@wits}{-}{-}{
144     \newcommand\pretty@wits{}
145   }{
146     \newcommand\pretty@wits{
147       \Bwith~
148       \evt@wits~
149     }
150   }

```

When there are no actions, SKIP is used. The keyword is changed depending on whether the event has parameters or not.

```

151   \B@ifstrequal{\evt@acts}{-}{-}{
152     \renewcommand\evt@acts{\SKIP}
153   }{
154     \newcommand\pretty@acts@tmp{
155       \evt@acts
156     }
157     \newcommand\pretty@acts@keyword{\Bthen}
158     \B@ifstrequal{\evt@pars}{-}{-}{
159       \B@ifstrequal{\evt@grds}{-}{-}{

```

```

160     \renewcommand\pretty@acts@keyword{\Bbegin}
161     }{}
162   }{}
163   \newcommand\pretty@acts{
164     \pretty@acts@keyword~
165     \pretty@acts@tmp~
166   }

```

Finally we put all the pretty-print pieces together.

```

167   \begin{array}{l}
168     \Bevt{\evt@label}\pretty@sts\pretty@absepts~\widehat{=}\sim
169     \pretty@pars
170     \pretty@grds
171     \pretty@wits
172     \pretty@acts
173   \Bend
174   \end{array}
175 } % END group
176 }

```

\B@makebox A wrapper macro to make a fbox with the boundary adjusted.

```

177 \newlength{\B@tmp@length}
178 \newcommand{\B@makebox}[1]{
179   {
180     \setlength{\B@tmp@length}{\fboxsep}
181     \setlength{\fboxsep}{2ex}
182     \fbox{#1}
183     \setlength{\fboxsep}{\B@tmp@length}
184   }
185 }

```

3.3 Declaration of Options for the Package

In this part various options for the package are defined.

3.3.1 Option for bounding boxes

By default, Event-B modelling elements, e.g., invariants, events, etc., are displayed in a bounding box. This `nobox` option enables them to be displayed without the bounding box.

\B@event Default definition displays Event-B events in a box.

```

186 \newcommand{\B@event}[7] [] {
187   \B@makebox{
188     \ensuremath{
189       \B@eventbase[#1]{#2}{#3}{#4}{#5}{#6}{#7}
190     }
191   }
192 }

```

\B@declaration Default definition displays Event-B declarations in a box.

```

193 \newcommand{\B@declaration}[2] {
194   \B@makebox{

```

```

195     \ensuremath{
196         \B@declarationbase{#1}{#2}
197     }
198 }
199 }

\B@section Default definition displays Event-B sections in a box
200 \newcommand{\B@section}[3] [] {
201     \B@makebox{
202         \ensuremath{
203             \B@sectionbase[#1]{#2}{#3}
204         }
205     }
206 }

Option “nobox” The above commands are redefined accordingly when option
nobox is enabled.
207 \DeclareOption{nobox}{

\B@event Redefine the definition without the bounding box.
208 \renewcommand{\B@event}[7] [] {
209     \B@eventbase[#1]{#2}{#3}{#4}{#5}{#6}{#7}
210 }

\B@declaration Redefine the definition without the bounding box.
211 \renewcommand{\B@declaration}[2] {
212     \B@declarationbase{#1}{#2}
213 }

\B@section Redefine the definition without the bounding box.
214 \renewcommand{\B@section}[3] [] {
215     \B@sectionbase[#1]{#2}{#3}
216 }
217 }

3.3.2 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.

\B@fontsize The font size used in the Bcode environment (defined later).
218 \newcommand{\B@fontsize}{\normalsize}

\B@vspace A vertical rule for spacing, defaulted to be 2ex.
219 \newcommand{\B@vspace}[1] [2ex]{\vspace{#1}}

\B@hspace A horizontal rule for spacing, defaulted to be 2em.
220 \newcommand{\B@hspace}[1] [2em]{\hspace{#1}}

\B@tab A small tab for spacing, defaulted to be \quad.
221 \newcommand{\B@tab}{\quad} % A small separation space

```

We subsequently redefined the above spacing commands when one of the options `small`, `compact`, `tiny` is enabled.

Option “small” For option `small` they are adjusted as follows.

```
222 \DeclareOption{small}{
\B@fontsize  Redefine to be \small for option small.
223   \renewcommand{\B@fontsize}{\small}

\B@vspace    Redefine to be 1ex for option small.
224   \renewcommand{\B@vspace}[1][1ex]{\ [#1]}

\B@hspace    Redefine to be 1em for option small.
225   \renewcommand{\B@hspace}[1][1em]{\hspace{#1}}

\B@tab       Redefine to be \ for option small.
226   \renewcommand{\B@tab}{\ }
227 }
```

Option “compact” For option `compact` the commands are adjusted as follows.

```
228 \DeclareOption{compact}{
\B@fontsize  Redefine to be \footnotesize for option compact.
229   \renewcommand{\B@fontsize}{\footnotesize}

\B@vspace    Redefine to be 0ex for option compact.
230   \renewcommand{\B@vspace}[1][0ex]{\ [#1]}

\B@hspace    Redefine to be 0.5em for option compact.
231   \renewcommand{\B@hspace}[1][0.5em]{\hspace{#1}}

\B@tab       Redefine to be \ for option compact.
232   \renewcommand{\B@tab}{\ }

Option nobox is enabled.
233   \ExecuteOptions{nobox}
234 }
```

Option “tiny” For option `tiny` the commands are adjusted as follows.

```
235 \DeclareOption{tiny}{
\B@fontsize  Redefine to be \scriptsize for option tiny.
236   \renewcommand{\B@fontsize}{\scriptsize}

\B@vspace    Redefine to be -0.5ex for option tiny.
237   \renewcommand{\B@vspace}[1][-0.5ex]{\ [#1]}

\B@hspace    Redefine to be 0.5em for option compact.
238   \renewcommand{\B@hspace}[1][0.5em]{\hspace{#1}}

\B@tab       Redefine to be \ for option compact.
239   \renewcommand{\B@tab}{\ }

Option nobox is enabled.
240   \ExecuteOptions{nobox}
241 }
```


3.3.3 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. We proceed with some definitions that can be redefined by these options.

```

\B@keyword Macro for Event-B keywords.
242 \newcommand{\B@keyword}[1]{\ensuremath{\B@keywordbase{#1}}\xspace}

\B@identifier Macro for Event-B identifiers.
243 \newcommand{\B@identifier}[1]{\ensuremath{\B@identifierbase{#1}}\xspace}

244 \newcommand{\B@label}[2][1]{\ensuremath{\B@label{#1}{#2}}\xspace}
245 \newcommand{\B@po}[1]{\ensuremath{\B@po{#1}}\xspace}
246 \DeclareOption{colour}{
247   \newcommand{\setBKeywordColour}[1]{\colorlet{B@keywordcolor}{#1}}
248   \setBKeywordColour{blue}
249   \newcommand{\setBIdentifierColour}[1]{\colorlet{B@identifiercolor}{#1}}
250   \setBIdentifierColour{blue!50!red}
251   \newcommand{\setBLabelColour}[1]{\colorlet{B@labelcolor}{#1}}
252   \setBLabelColour{green!50!black}
253   \newcommand{\setBPOLColour}[1]{\colorlet{B@pocolor}{#1}}
254   \setBPOLColour{red}
255   \renewcommand{\B@keyword}[1]{
256     \ensuremath{\textcolor{B@keywordcolor}{\B@keywordbase{#1}}}\xspace
257   }
258   \renewcommand{\B@identifier}[1]{
259     \ensuremath{\textcolor{B@identifiercolor}{\B@identifierbase{#1}}}\xspace
260   }
261   \renewcommand{\B@label}[2][1]{
262     \ensuremath{\textcolor{B@labelcolor}{\B@label{#1}{#2}}}\xspace
263   }
264   \renewcommand{\B@po}[1]{
265     \ensuremath{\textcolor{B@pocolor}{\B@po{#1}}}\xspace
266   }
267 }
268 \DeclareOption{color}{
269   \ExecuteOptions{colour}
270 }
271
```

After declaration of options, we execute them accordingly.

```
272 \ProcessOptions
```

3.4 Commands for Pretty-Print Event-B Models

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

```
273 \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.

```
274 \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($\backslash\backslash$)-separated list of elements.

```

275 \newcommand{\carriersets}[1]{
276   \B@declaration{sets}{#1}
277 }

278 \newcommand{\constants}[1]{
279   \B@declaration{constants}{#1}
280 }

281 \newcommand{\axioms}[2][]{
282   \B@section[#1]{axioms}{#2}
283 }

284 \newcommand{\variables}[1]{
285   \B@declaration{variables}{#1}
286 }

287 \newcommand{\invariants}[2][]{
288   \B@section[#1]{invariants}{#2}
289 }

290 \newcommand{\variant}[1]{
291   \B@declaration{variant}{#1}
292 }

```

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

```

293 \newcommand{\Bany}{\B@keyword{any}}
294 \newcommand{\Bbegin}{\B@keyword{begin}}
295 \newcommand{\Bend}{\B@keyword{end}}
296 \newcommand{\Brefines}{\B@keyword{refines}}
297 \newcommand{\Bstatus}{\B@keyword{status}}
298 \newcommand{\Bthen}{\B@keyword{then}}
299 \newcommand{\Bwhen}{\B@keyword{when}}
300 \newcommand{\Bwhere}{\B@keyword{where}}
301 \newcommand{\Bwith}{\B@keyword{with}}

```

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

```

302 \newcommand{\Bctx}[1]{\ensuremath{\mathbf{#1}}\xspace}
303 \newcommand{\Bset}[1]{\Bidentifier{#1}}
304 \newcommand{\Bcst}[1]{\Bidentifier{#1}}
305 \newcommand{\Baxm}[1]{\Blabel{#1}}
306 \newcommand{\Bthm}[1]{\Blabel{thm}{#1}}
307
308 \newcommand{\Bmch}[1]{\ensuremath{\mathbf{#1}}\xspace}
309 \newcommand{\Bvrb}[1]{\Bidentifier{#1}}
310 \newcommand{\Binv}[1]{\Blabel{#1}}
311 \newcommand{\Bvt}[1]{\Blabel{#1}}
312 \newcommand{\Bpar}[1]{\Bidentifier{#1}}
313 \newcommand{\Bact}[1]{\Blabel{#1}}
314 \newcommand{\Bgrd}[1]{\Blabel{#1}}
315 \newcommand{\Bbap}[1]{\hbox{\sl\bfseries #1}}

```

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

```

316 \newcommand{\B@newmacro}[3][]{
317   \ifstrequal{#1}{-}{
318     \expandafter\def\csname #2\endcsname{#3{#2}}
319   }{
320     \expandafter\def\csname #1\endcsname{#3{#2}}
321   }
322 }

323 \newcommand{\newBctx}[2][]{\B@newmacro[#1]{#2}{\Bctx}}
324 \newcommand{\newBset}[2][]{\B@newmacro[#1]{#2}{\Bset}}
325 \newcommand{\newBcst}[2][]{\B@newmacro[#1]{#2}{\Bcst}}
326 \newcommand{\newBaxm}[2][]{\B@newmacro[#1]{#2}{\Baxm}}
327 \newcommand{\newBthm}[2][]{\B@newmacro[#1]{#2}{\Bthm}}
328 \newcommand{\newBmch}[2][]{\B@newmacro[#1]{#2}{\Bmch}}
329 \newcommand{\newBvrb}[2][]{\B@newmacro[#1]{#2}{\Bvrb}}
330 \newcommand{\newBinvt}[2][]{\B@newmacro[#1]{#2}{\Binvt}}
331 \newcommand{\newBevt}[2][]{\B@newmacro[#1]{#2}{\Bevt}}
332 \newcommand{\newBpar}[2][]{\B@newmacro[#1]{#2}{\Bpar}}
333 \newcommand{\newBgrd}[2][]{\B@newmacro[#1]{#2}{\Bgrd}}
334 \newcommand{\newBact}[2][]{\B@newmacro[#1]{#2}{\Bact}}

335
336 %%%% Theorem Proof Obligation
337 %%%% Print the theorem proof obligation, given the theorem label.
338 %%%% Arguments:
339 %%%% 1. Theorem label
340 %%%%
341 %%%% Usage:
342 %%%% - \thmpo{thm} will produce "thm/THM"
343 \newcommand{\thmpo}[1]{\Bthm{#1}/\Bpo{THM}}
344
345 %%%% Axiom Well-definedness Proof Obligation
346 %%%% Print the axiom well-definedness proof obligation, given the
347 %%%% axiom label.
348 %%%% Arguments:
349 %%%% 1. Axiom label
350 %%%%
351 %%%% Usage:
352 %%%% - \axmwdpo{axm} will produce "axm/WD"
353 \newcommand{\axmwdpo}[1]{\Baxm{#1}/\Bpo{WD}}
354
355 %%%% Invariant Proof Obligation
356 %%%% Print the invariant proof obligation, given the event name and
357 %%%% invariant label
358 %%%% Arguments:
359 %%%% 1. Event name
360 %%%% 2. Invariant label

```

```

361 %%%%
362 %%%% Usage:
363 %%%% - \invpo{evt}{inv} will produce "evt/inv/INV"
364 \newcommand{\invpo}[2]{\Bevt{#1}/\Binv{#2}/\Bpo{INV}}
365
366 %%%% Theorem (in guard) Proof Obligation
367 %%%% Print the simulation proof obligation, given the event name and
368 %%%% the theorem (in guard) label.
369 %%%% Arguments:
370 %%%% 1. Event name
371 %%%% 2. Theorem (in guard) label
372 %%%%
373 %%%% Usage:
374 %%%% - \grdthmpo{evt}{thm} will produce "evt/thm/THM"
375 \newcommand{\grdthmpo}[2]{\Bevt{#1}/\Bthm{#2}/\Bpo{THM}}
376
377 %%%% Feasibility Proof Obligation
378 %%%% Print the feasibility proof obligation, given the event name and
379 %%%% the action label
380 %%%% Arguments:
381 %%%% 1. Event name
382 %%%% 2. Action label
383 %%%%
384 %%%% Usage:
385 %%%% - \fispo{evt}{act} will produce "evt/act/FIS"
386 \newcommand{\fispo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{FIS}}
387
388 %%%% Variant finiteness Proof Obligation
389 %%%% Print the Variant finiteness proof obligation
390 %%%% Arguments: No arguments
391 %%%%
392 %%%% Usage:
393 %%%% - \finpo will produce "FIN"
394 \newcommand{\finpo}{\Bpo{FIN}}
395
396 %%%% Variant Proof Obligation
397 %%%% Print the guard strengthen proof obligation, given the event name
398 %%%% Arguments:
399 %%%% 1. Event name
400 %%%%
401 %%%% Usage:
402 %%%% - \grdpo{evt} will produce "evt/VAR"
403 \newcommand{\varpo}[1]{\Bevt{#1}/\Bpo{VAR}}
404
405 %%%% Simulation Proof Obligation
406 %%%% Print the simulation proof obligation, given the event name and
407 %%%% the action label.
408 %%%% Arguments:
409 %%%% 1. Event name
410 %%%% 2. Action label
411 %%%%
412 %%%% Usage:

```

```

413 %%%% - \simpo{evt}{act} will produce "evt/act/SIM"
414 \newcommand{\simpo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{SIM}}
415
416 %%%% Guard Strengthen Proof Obligation
417 %%%% Print the guard strengthen proof obligation, given the event
418 %%%% name and the guard label
419 %%%% Arguments:
420 %%%% 1. (Abstract) Event name
421 %%%% 2. (Abstract) Guard label
422 %%%%
423 %%%% Usage:
424 %%%% - \grdpo{evt}{grd} will produce "evt/grd/GRD"
425 \newcommand{\grdpo}[2]{\Bevt{#1}/\Bgrd{#2}/\Bpo{GRD}}
426
427 %%%% Variant Natural Number Proof Obligation
428 %%%% Print the Variant Natural Number proof obligation, given the event name
429 %%%% Arguments:
430 %%%% 1. Event name
431 %%%%
432 %%%% Usage:
433 %%%% - \natpo{evt} will produce "evt/NAT"
434 \newcommand{\natpo}[1]{\Bevt{#1}/\Bpo{NAT}}
435

```

\inlineevent

```

436 \newcommand{\inlineevent}[7][]{
437   \B@inlineeventbase{#1}{#2}{#3}{#4}{#5}{#6}{#7}
438 }

439 \newcommand{\B@label}[2][]{
440   \ifstrequal{#1}{}{
441     \mathsf{#2}
442   }{
443     \mathit{#2}
444   }
445 }
446
447
448
449
450
451
452 \newcommand{\B@po}[1]{\ensuremath{\mathsf{#1}}\xspace}
453
454 %%%% (BEGIN) Macros for Pretty-Print Event-B Components %%%
455 \newcommand{\SKIP}{\textsc{skip}\xspace}
456 %

```

\INITIALISATION

```

457 %%%% INITIALISATION label
458 \newBevt{INITIALISATION}
459

```

```

460 %%% Pretty print the initialisation: no ‘refines’ clause. no parameters, no
461 %%% guards
462 %%% Arguments:
463 %%% 1. (Newline(\)-separated) list of assignments.
464 %%%
465 %%% Usage: \initialisation{S1(v,x,y)\S2(w,x,y)}
466 %%% will produce the following
467 %%%
468 %%% init
469 %%% begin
470 %%% S1(v, x, y)
471 %%% S2(w, x, y)
472 %%% end
473 %%%
474 \newcommand{\initialisation}[1]{
475   \event{\INITIALISATION}{-}{-}{-}{#1}
476 }
477
478 %\newcommand{\event}[7][[]]{
479 %  \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
480 %}
481
482 \let\event\B@event
483 \let\Bvspace\B@vspace
484 \let\Bhspace\B@hspace
```

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.

	\B@identifierbase .	\B@sectionbase
_ 226, 232, 239 <u>5</u> , 243, 259 <u>11</u> , 203, 215
	\B@ifstrequal	\B@tab 7, 15,
A	. <u>25</u> , 35, 40, 45,	20, 38, 43, 48,
\axioms 281	50, 58, 67, 77,	54, 60, 71, 72,
\axmwdpo 352, 353	86, 87, 116, 121,	81, 85, 88, 103,
	126, 131, 137,	<u>221</u> , <u>226</u> , <u>232</u> , <u>239</u>
B	143, 151, 158, 159	\B@tmp@length
\B@declaration	\B@inlineeventbase 177, 180, 183
..... <u>193</u> , <u>211</u> , <u>107</u> , 437	\B@vspace <u>219</u> ,
276, 279, 285, 291	\B@keyword .. 8, 14,	<u>224</u> , <u>230</u> , <u>237</u> , 483
\B@declarationbase .	<u>242</u> , 255, 293–301	\Bact . 313, 334, 386, 414
..... <u>6</u> , 196, 212	\B@keywordbase	\Bany 48, 129, 293
\B@event <u>4</u> , 242, 256	\Baxm 305, 326, 353
. <u>186</u> , <u>208</u> , 479, 482	\B@label .. 244, 262, 439	\Bbap 315
\B@eventbase <u>26</u> , 189, 209	\B@makebox	\Bbegin ... 88, 160, 294
\B@fontsize ... <u>218</u> ,	. <u>177</u> , 187, 194, 201	\Bcst 304, 325
<u>223</u> , <u>229</u> , <u>236</u> , 274	\B@newmacro 316, 323–334	\Bctx 302, 323
\B@hspace <u>220</u> ,	\B@po 245, 265, 452	\Bend 103, 173, 295
<u>225</u> , <u>231</u> , <u>238</u> , 484	\B@section	\Bevt 96, 168, 311, 331,
\B@identifier 243	. 200, 214, 282, 288	364, 375, 386,

403, 414, 425, 434	\evt@label	P
\bfseries 315	. . . 29, 96, 110, 168	\pretty@absevt 41,
\Bgrd 314, 333, 425	\evt@pars . . . 31, 45,	43, 98, 122, 124, 168
\Bhspace 484	48, 58, 86, 112,	\pretty@acts
\Bidentifier 243, 258,	126, 129, 137, 158	. . . 91, 102, 163, 172
303, 304, 309, 312	\evt@sts 28,	\pretty@acts@keyword
\Binv 310, 330, 364	35, 38, 109, 116, 119 85,
\Blabel 244,	\evt@wits 33,	88, 92, 157, 160, 164
261, 305, 306,	67, 73, 114, 143, 148	\pretty@acts@tmp . .
310, 311, 313, 314		. . . 80, 93, 154, 165
\Bmch 308, 328	F	\pretty@grds
\Bpar 312, 332	\finpo 393, 394	. . . 51, 59, 64,
\Bpo 245, 264,	\fispo 385, 386	100, 132, 140, 170
343, 353, 364,		\pretty@grds@tmp 53,
375, 386, 394,	G	61, 64, 134, 138, 140
403, 414, 425, 434	\grdpo 402, 424, 425	\pretty@pars . . . 46,
\Brefines . . 43, 124, 296	\grdthmpo 374, 375	48, 99, 127, 129, 169
\Bset 303, 324		\pretty@sts 36,
\Bstatus 38, 297	I	38, 97, 117, 119, 168
\Bthen 85, 157, 298	\INITIALISATION 457, 475	\pretty@wits . 68, 70,
\Bthm . . 306, 327, 343, 375	\initialisation 465, 474	101, 144, 146, 171
\Bvrb 309, 329	\inlineevent 436	
\Bvspace 483	\invariants 287	S
\Bwhen 60, 138, 299	\invpo 363, 364	\setBIdentifierColour
\Bwhere . . . 48, 129, 300	 249, 250
\Bwith 71, 147, 301	L	\setBKeywordColour .
	\let 482–484 247, 248
C		\setBLabelColour . .
\carriersets 275	N 251, 252
\constants 278	\natpo 433, 434	\setBPOColour . . 253, 254
	\newBact 334	\simpo 413, 414
E	\newBaxm 326	\SKIP 78, 152, 455
\event 475, 478, 482	\newBcst 325	\sl 315
\eventB 273	\newBctx 323	
\evt@absevt 30,	\newBevt 331, 458	T
40, 43, 111, 121, 124	\newBgrd 333	\thmpo 342, 343
\evt@acts	\newBinv 330	
. . 34, 77, 78, 82,	\newBmch 328	V
115, 151, 152, 155	\newBpar 332	\variables 284
\evt@grds	\newBset 324	\variant 290
. . 32, 50, 55, 87,	\newBthm 327	\varpo 403
113, 131, 135, 159	\newBvrb 329	

Change History

v1.0	v1.1
General: Initial version 1	General: Re-implement how op-
	tions are defined, added options
v1.0.1	'nobox' 1
\B@declaration: Ensure math-	v1.1.1
mode 6	General: Updated documentation . . 1
\B@event: Ensure math-mode 6	v2.0
\B@section: Ensure math-mode . . 7	General: Major re-implementation,

use etoolbox instead of ifthen	1	<code>\INITIALISATION</code> : Renamed from	
Updated documentation, added		init	13
DoNotIndex	1	<code>\inlineevent</code> : Renamed from	
<code>\B@makebox</code> : Added	6	eventinline	13