

# 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Usage</b>	<b>1</b>
2.1	Package Options . . . . .	1
<b>3</b>	<b>Implementation</b>	<b>2</b>
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  $\text{\LaTeX}$ .

## 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`.

```
1 \RequirePackage{xspace}
2 \RequirePackage{xcolor}
3 \RequirePackage{ifthen}
4 \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.

```
5 \newlength{\B@oldfboxsep}
6 \newcommand{\event}[7][]{
7   \setlength{\B@oldfboxsep}{\fboxsep}
8   \setlength{\fboxsep}{2ex}
9   \fbox{
10     \ensuremath{
11       \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
12     }
13   }
14   \setlength{\fboxsep}{\B@oldfboxsep}
15 }
16
17 \newcommand{\B@declaration}[2]{
18   \setlength{\B@oldfboxsep}{\fboxsep}
19   \setlength{\fboxsep}{2ex}
20   \fbox{
21     \ensuremath{
22       \B@declarationbase[#1]{#2}
23     }
24   }
25   \setlength{\fboxsep}{\B@oldfboxsep}
26 }
27
28 \newcommand{\B@section}[3][]{
29   \setlength{\B@oldfboxsep}{\fboxsep}
30   \setlength{\fboxsep}{2ex}
31   \fbox{
32     \ensuremath{
33       \B@sectionbase[#1]{#2}{#3}
34     }
35   }
36   \setlength{\fboxsep}{\B@oldfboxsep}
37 }
38
39 \DeclareOption{nobox}{
40   \renewcommand{\event}[7][]{
```

```

41 \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
42 }
43
44 \renewcommand{\B@declaration}[2]{
45 \B@declarationbase{#1}{#2}
46 }
47
48 \renewcommand{\B@section}[3][]{
49 \B@sectionbase{#1}{#2}{#3}
50 }
51 }
52

```

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

```

53 \newcommand{\B@fontsize}{\normalsize} % The font size used in Bcode environment
54 \newcommand{\Bvspace}[1][2ex]{\setlength{\B@vspace}{#1}} % Vertical space
55 \newcommand{\Bhspace}[1][2em]{\setlength{\B@hspace}{#1}} % Horizontal space
56 \newcommand{\B@tab}{\quad} % A small separation space
57
58 \DeclareOption{small}{
59 \renewcommand{\B@fontsize}{\small}
60 \renewcommand{\Bvspace}[1][1ex]{\setlength{\B@vspace}{#1}}
61 \renewcommand{\Bhspace}[1][1em]{\setlength{\B@hspace}{#1}}
62 \renewcommand{\B@tab}{\ }
63 }
64 \DeclareOption{compact}{
65 \renewcommand{\B@fontsize}{\footnotesize}
66 \renewcommand{\Bvspace}[1][0ex]{\setlength{\B@vspace}{#1}}
67 \renewcommand{\Bhspace}[1][0.5em]{\setlength{\B@hspace}{#1}}
68 \renewcommand{\B@tab}{\ }
69 \ExecuteOptions{nobox}
70 }
71 \DeclareOption{tiny}{
72 \renewcommand{\B@fontsize}{\scriptsize}
73 \renewcommand{\Bvspace}[1][-0.5ex]{\setlength{\B@vspace}{#1}}
74 \renewcommand{\Bhspace}[1][0.5em]{\setlength{\B@hspace}{#1}}
75 \renewcommand{\B@tab}{\ }
76 \ExecuteOptions{nobox}
77 }
78

```

**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.

```

79 \newcommand{\B@keyword}[1]{\ensuremath{\B@keywordbase{#1}}\xspace}
80 \newcommand{\B@identifier}[1]{\ensuremath{\B@identifier{#1}}\xspace}
81 \newcommand{\B@label}[2][]{\ensuremath{\B@label{#1}{#2}}\xspace}
82 \newcommand{\B@po}[1]{\ensuremath{\B@po{#1}}\xspace}

```

```

83 \DeclareOption{colour}{
84   \newcommand{\setBKeywordColour}[1]{\colorlet{B@keywordcolor}{#1}}
85   \setBKeywordColour{blue}
86   \newcommand{\setBIdentifierColour}[1]{\colorlet{B@identifiercolor}{#1}}
87   \setBIdentifierColour{blue!50!red}
88   \newcommand{\setBLabelColour}[1]{\colorlet{B@labelcolor}{#1}}
89   \setBLabelColour{green!50!black}
90   \newcommand{\setBP0Colour}[1]{\colorlet{B@pocolor}{#1}}
91   \setBP0Colour{red}
92   \renewcommand{\B@keyword}[1]{
93     \ensuremath{\textcolor{B@keywordcolor}{\B@keywordbase{#1}}}\xspace
94   }
95   \renewcommand{\B@identifier}[1]{
96     \ensuremath{\textcolor{B@identifiercolor}{\B@identifier{#1}}}\xspace
97   }
98   \renewcommand{\B@label}[2][]{
99     \ensuremath{\textcolor{B@labelcolor}{\B@label{#1}{#2}}}\xspace
100  }
101  \renewcommand{\B@po}[1]{
102    \ensuremath{\textcolor{B@pocolor}{\B@po{#1}}}\xspace
103  }
104 }
105 \DeclareOption{color}{
106   \ExecuteOptions{colour}
107 }
108

```

After declaration of options, we execute them accordingly.

```

109 \ProcessOptions

```

### 3.3 Commands for Pretty-Print Event-B Models

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

```

110 \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.

```

111 \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.

```

112 \newcommand{\carriersets}[1]{
113   \B@declaration{sets}{#1}
114 }
115 \newcommand{\constants}[1]{
116   \B@declaration{constants}{#1}
117 }
118 \newcommand{\axioms}[2][]{
119   \B@section{#1}{axioms}{#2}
120 }

```

```

121 \newcommand{\variables}[1]{
122   \B@declaration{variables}{#1}
123 }

124 \newcommand{\invariants}[2][]{
125   \B@section[#1]{invariants}{#2}
126 }

127 \newcommand{\variant}[1]{
128   \B@declaration{variant}{#1}
129 }

```

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

```

130 \newcommand{\Bany}{\B@keyword{any}}
131 \newcommand{\Bbegin}{\B@keyword{begin}}
132 \newcommand{\Bend}{\B@keyword{end}}
133 \newcommand{\Brefines}{\B@keyword{refines}}
134 \newcommand{\Bstatus}{\B@keyword{status}}
135 \newcommand{\Bthen}{\B@keyword{then}}
136 \newcommand{\Bwhen}{\B@keyword{when}}
137 \newcommand{\Bwhere}{\B@keyword{where}}
138 \newcommand{\Bwith}{\B@keyword{with}}

```

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

```

139 \newcommand{\Bctx}[1]{\ensuremath{\mathbf{#1}}\xspace}
140 \newcommand{\Bset}[1]{\Bidentifier{#1}}
141 \newcommand{\Bcst}[1]{\Bidentifier{#1}}
142 \newcommand{\Baxm}[1]{\Blabel{#1}}
143 \newcommand{\Bthm}[1]{\Blabel{thm}{#1}}
144
145 \newcommand{\Bmch}[1]{\ensuremath{\mathbf{#1}}\xspace}
146 \newcommand{\Bvrb}[1]{\Bidentifier{#1}}
147 \newcommand{\Binv}[1]{\Blabel{#1}}
148 \newcommand{\Bevt}[1]{\Blabel{#1}}
149 \newcommand{\Bpar}[1]{\Bidentifier{#1}}
150 \newcommand{\Bact}[1]{\Blabel{#1}}
151 \newcommand{\Bgrd}[1]{\Blabel{#1}}
152 \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.

```

153 \newcommand{\B@newmacro}[3][]{
154   \ifstrequal{#1}{\B@newmacro}{#2}{\B@newmacro}{#3}{#2}}
155   \expandafter\def\csname #2\endcsname{#3{#2}}
156 }{
157   \expandafter\def\csname #1\endcsname{#3{#2}}
158 }
159 }

160 \newcommand{\newBctx}[2][]{\B@newmacro{#1}{#2}{\Bctx}}
161 \newcommand{\newBset}[2][]{\B@newmacro{#1}{#2}{\Bset}}
162 \newcommand{\newBcst}[2][]{\B@newmacro{#1}{#2}{\Bcst}}

```

```

163 \newcommand{\newBaxm}[2] [] {\B@newmacro[#1]{#2}{\Baxm}}
164 \newcommand{\newBthm}[2] [] {\B@newmacro[#1]{#2}{\Bthm}}
165 \newcommand{\newBmch}[2] [] {\B@newmacro[#1]{#2}{\Bmch}}
166 \newcommand{\newBvrb}[2] [] {\B@newmacro[#1]{#2}{\Bvrb}}
167 \newcommand{\newBinvt}[2] [] {\B@newmacro[#1]{#2}{\Binvt}}
168 \newcommand{\newBevt}[2] [] {\B@newmacro[#1]{#2}{\Bevt}}
169 \newcommand{\newBpar}[2] [] {\B@newmacro[#1]{#2}{\Bpar}}
170 \newcommand{\newBgrd}[2] [] {\B@newmacro[#1]{#2}{\Bgrd}}
171 \newcommand{\newBact}[2] [] {\B@newmacro[#1]{#2}{\Bact}}
172
173 %%%% Theorem Proof Obligation
174 %%%% Print the theorem proof obligation, given the theorem label.
175 %%%% Arguments:
176 %%%% 1. Theorem label
177 %%%%
178 %%%% Usage:
179 %%%% - \thmpo{thm} will produce "thm/THM"
180 \newcommand{\thmpo}[1]{\Bthm[#1]/\Bpo{THM}}
181
182 %%%% Axiom Well-definedness Proof Obligation
183 %%%% Print the axiom well-definedness proof obligation, given the
184 %%%% axiom label.
185 %%%% Arguments:
186 %%%% 1. Axiom label
187 %%%%
188 %%%% Usage:
189 %%%% - \axmwdpo{axm} will produce "axm/WD"
190 \newcommand{\axmwdpo}[1]{\Baxm[#1]/\Bpo{WD}}
191
192 %%%% Invariant Proof Obligation
193 %%%% Print the invariant proof obligation, given the event name and
194 %%%% invariant label
195 %%%% Arguments:
196 %%%% 1. Event name
197 %%%% 2. Invariant label
198 %%%%
199 %%%% Usage:
200 %%%% - \invpo{evt}{inv} will produce "evt/inv/INV"
201 \newcommand{\invpo}[2]{\Bevt[#1]/\Binvt[#2]/\Bpo{INV}}
202
203 %%%% Theorem (in guard) Proof Obligation
204 %%%% Print the simulation proof obligation, given the event name and
205 %%%% the theorem (in guard) label.
206 %%%% Arguments:
207 %%%% 1. Event name
208 %%%% 2. Theorem (in guard) label
209 %%%%
210 %%%% Usage:
211 %%%% - \grdthmpo{evt}{thm} will produce "evt/thm/THM"

```

```

212 \newcommand{\grdthmpo}[2]{\Bevt{#1}/\Bthm{#2}/\Bpo{THM}}
213
214 %%%% Feasibility Proof Obligation
215 %%%% Print the feasibility proof obligation, given the event name and
216 %%%% the action label
217 %%%% Arguments:
218 %%%% 1. Event name
219 %%%% 2. Action label
220 %%%%
221 %%%% Usage:
222 %%%% - \fispo{evt}{act} will produce "evt/act/FIS"
223 \newcommand{\fispo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{FIS}}
224
225 %%%% Variant finiteness Proof Obligation
226 %%%% Print the Variant finiteness proof obligation
227 %%%% Arguments: No arguments
228 %%%%
229 %%%% Usage:
230 %%%% - \finpo will produce "FIN"
231 \newcommand{\finpo}{\Bpo{FIN}}
232
233 %%%% Variant Proof Obligation
234 %%%% Print the guard strengthen proof obligation, given the event name
235 %%%% Arguments:
236 %%%% 1. Event name
237 %%%%
238 %%%% Usage:
239 %%%% - \grdpo{evt} will produce "evt/VAR"
240 \newcommand{\varpo}[1]{\Bevt{#1}/\Bpo{VAR}}
241
242 %%%% Simulation Proof Obligation
243 %%%% Print the simulation proof obligation, given the event name and
244 %%%% the action label.
245 %%%% Arguments:
246 %%%% 1. Event name
247 %%%% 2. Action label
248 %%%%
249 %%%% Usage:
250 %%%% - \simpo{evt}{act} will produce "evt/act/SIM"
251 \newcommand{\simpo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{SIM}}
252
253 %%%% Guard Strengthen Proof Obligation
254 %%%% Print the guard strengthen proof obligation, given the event
255 %%%% name and the guard label
256 %%%% Arguments:
257 %%%% 1. (Abstract) Event name
258 %%%% 2. (Abstract) Guard label
259 %%%%
260 %%%% Usage:
261 %%%% - \grdpo{evt}{grd} will produce "evt/grd/GRD"
262 \newcommand{\grdpo}[2]{\Bevt{#1}/\Bgrd{#2}/\Bpo{GRD}}
263

```

```

264 %%%% Variant Natural Number Proof Obligation
265 %%%% Print the Variant Natural Number proof obligation, given the event name
266 %%%% Arguments:
267 %%%% 1. Event name
268 %%%%
269 %%%% Usage:
270 %%%% - \natpo{evt} will produce "evt/NAT"
271 \newcommand{\natpo}[1]{\Bevt{#1}/\Bpo{NAT}}
272
273 \newcommand{\B@keywordbase}[1]{\mathbf{#1}}
274 \newcommand{\B@identifier}[1]{\mathit{#1}}
275 \newcommand{\B@label}[2][]{
276   \ifstrequal{#1}{}{
277     \mathsf{#2}
278   }{
279     \mathit{#2}
280   }
281 }
282
283
284
285 \newcommand{\eventinline}[7][]{
286   \setlength{\B@oldfboxsep}{\fboxsep}
287   \setlength{\fboxsep}{2ex}
288   \fbox{
289     \ensuremath{
290       \B@eventinline[#1]{#2}{#3}{#4}{#5}{#6}{#7}
291     }
292   }
293   \setlength{\fboxsep}{\B@oldfboxsep}
294 }
295
296
297 \newcommand{\B@declarationbase}[2]{
298   \begin{array}{l@{\B@tab}l}
299     \B@keyword{#1:} & #2
300   \end{array}
301 }
302
303 \newcommand{\B@sectionbase}[3][]{
304   \ifstrequal{#1}{}{
305     \begin{array}{l}
306       \B@keyword{#2:} \\
307       \begin{array}{l@{\B@tab}l}
308         #3
309       \end{array}
310     \end{array}
311   }{
312     \begin{array}{l@{\B@tab}l}
313       #3
314     \end{array}
315   }
316 }
317

```



```

318 \newcommand{\B@po}[1]{\ensuremath{\mathsf{\#1}}\xspace}
319
320 %%%% (BEGIN) Macros for Pretty-Print Event-B Components %%
321 \newcommand{\SKIP}{\textsc{skip}}
322
323
324 %%%% Pretty print an general Event-B event
325 %%%% Arguments:
326 %%%% 1. (Optional) convergence status.
327 %%%% 2. Name of the event.
328 %%%% 3. Name of the abstract event.
329 %%%% 4. (Comma-separated) list of parameters.
330 %%%% 5. (Newline\\)-separated) list of guards.
331 %%%% 6. (Newline\\)-separated) list of witness predicates.
332 %%%% 7. (Newline\\)-separated) list of assignments.
333 %%%%
334 %%%% Usage: \B@event[conv]{conc}{abs}{x,y}{G1(x,y)\G2(x,y)}{W1\\W2}{S1(v,x,y)\S2(w,x,y)}
335 %%%% will produce the following
336 %%%%
337 %%%% conc
338 %%%% refines abs
339 %%%% status conv
340 %%%% any x, y where
341 %%%% G1(x, y)
342 %%%% G2(x, y)
343 %%%% with
344 %%%% W1
345 %%%% W2
346 %%%% then
347 %%%% S1(v, x, y)
348 %%%% S2(w, x, y)
349 %%%% end
350 %%%%
351 %%%% Special case:
352 %%%% - Empty abstract event --> refines clause is omitted.
353 %%%% - Empty convergence status --> status clause is omitted.
354 %%%% - Empty witness --> with clause is omitted.
355 %%%% - Empty parameters, empty guards --> begin ... end
356 %%%% - Empty parameters --> when ... then ... end
357 %%%% - Empty actions --> \SKIP
358 \newcommand{\B@ifstrequal}{\expandafter\ifstrequal\expandafter}
359 \newcommand{\B@event}[7][]{%
360 { % BEGIN group
361 \newcommand\evt@sts{\#1}% Event status
362 \newcommand\evt@label{\#2}% Event label
363 \newcommand\evt@absevt{\#3}% Abstract event
364 \newcommand\evt@pars{\#4}% Event parameters
365 \newcommand\evt@grds{\#5}% Event guards
366 \newcommand\evt@wits{\#6}% Event witnesses
367 \newcommand\evt@acts{\#7}% Event actions
368 %% Pretty-print convergence status
369 \B@ifstrequal{\evt@sts}{%
370 \newcommand\pretty@sts{}

```

```

371   }{
372     \newcommand\pretty@sts{\B@tab\Bstatus \B@tab \evt@sts \\\}
373   }
374   % Pretty-print abstract events
375   \B@ifstrequal{\evt@absevt}{-}{
376     \newcommand\pretty@absevt{
377   }{
378     \newcommand\pretty@absevt{\B@tab\Brefines \B@tab \evt@absevt} \\\}
379   }
380   % Pretty-print parameters
381   \B@ifstrequal{\evt@pars}{-}{
382     \newcommand\pretty@pars{
383   }{
384     \newcommand\pretty@pars{\B@tab\Bany \B@tab \evt@pars \B@tab \Bwhere \\\}
385   }
386   % Pretty-print guards
387   \B@ifstrequal{\evt@grds}{-}{
388     \newcommand\pretty@grds{
389   }{
390     \newcommand\pretty@grds@tmp{
391       \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
392         \evt@grds
393       \end{array}\\\
394     }
395     \B@ifstrequal{\evt@pars}{-}{
396       \newcommand\pretty@grds{
397         \B@tab \Bwhen \\\
398         \pretty@grds@tmp
399       }
400     }{
401       \newcommand\pretty@grds{\pretty@grds@tmp}
402     }
403   }
404   % Pretty-print witnesses
405   \B@ifstrequal{\evt@wits}{-}{
406     \newcommand\pretty@wits{
407   }{
408     \newcommand\pretty@wits{
409       \B@tab\Bwith\\\
410       \begin{array}{@{\B@tab\B@tab}ll}
411         \evt@wits
412       \end{array}\\\
413     }
414   }
415   % Pretty-print actions
416   \B@ifstrequal{\evt@acts}{-}{
417     \renewcommand\evt@acts{\SKIP}
418   }{}
419   \newcommand\pretty@acts@tmp{
420     \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
421       \evt@acts
422     \end{array}\\\
423   }
424   \newcommand\pretty@acts@keyword{\B@tab\Bthen \\\}

```

```

425 \B@ifstrequal{\evt@pars}{-}{-}{
426 \B@ifstrequal{\evt@grds}{-}{-}{
427 \renewcommand\pretty@acts@keyword{\B@tab\Bbegin \\\}
428 }{-}{
429 }{-}{
430 \newcommand\pretty@acts{
431 \pretty@acts@keyword
432 \pretty@acts@tmp
433 }
434 % Really do it now
435 \begin{array}{l}
436 \Bvt{\evt@label} \\\
437 \pretty@sts
438 \pretty@absevs
439 \pretty@pars
440 \pretty@grds
441 \pretty@wits
442 \pretty@acts
443 \B@tab\Bend
444 \end{array}
445 } % END group
446 }
447
448 %%% Pretty print an general Event-B event
449 %%% Arguments:
450 %%% 1. (Optional) convergence status.
451 %%% 2. Name of the event.
452 %%% 3. Name of the abstract event.
453 %%% 4. (Comma-separated) list of parameters.
454 %%% 5. (\land-separated) list of guards.
455 %%% 6. (\land-separated) list of witness predicates.
456 %%% 7. (||-separated) list of assignments.
457 %%%
458 %%% Usage: \B@eventinline[conv]{conc}{abs}{x,y}{G1 \land G2}{W1 \land W2}{S1 || S2}
459 %%% will produce the following
460 %%%
461 %%% conc (conv) refines abs = any x,y where G1 \land G2 with W1 \land W2 then S1 || S2 end
462 %%%
463 %%% Special case:
464 %%% - Empty abstract event --> refines clause is omitted.
465 %%% - Empty convergence status --> status clause is omitted.
466 %%% - Empty witness --> with clause is omitted.
467 %%% - Empty parameters, empty guards --> begin ... end
468 %%% - Empty parameters --> when ... then ... end
469 %%% - Empty actions --> \SKIP
470 \newcommand{\B@eventinline}[7][]{
471 { % BEGIN group
472 \newcommand\evt@sts{#1}% Event status
473 \newcommand\evt@label{#2}% Event label
474 \newcommand\evt@absevs{#3}% Abstract event
475 \newcommand\evt@pars{#4}% Event parameters
476 \newcommand\evt@grds{#5}% Event guards
477 \newcommand\evt@wits{#6}% Event witnesses
478 \newcommand\evt@acts{#7}% Event actions

```

```

479 %% Pretty-print convergence status
480 \B@ifstrequal{\evt@sts}{}{
481   \newcommand\pretty@sts{
482 }{
483   \newcommand\pretty@sts{(\evt@sts)}
484 }
485 % Pretty-print abstract events
486 \B@ifstrequal{\evt@absevt}{}{
487   \newcommand\pretty@absevt{
488 }{
489   \newcommand\pretty@absevt{~\Brefines~\evt@absevt}
490 }
491 % Pretty-print parameters
492 \B@ifstrequal{\evt@pars}{}{
493   \newcommand\pretty@pars{
494 }{
495   \newcommand\pretty@pars{\Bany~\evt@pars~\Bwhere~}
496 }
497 % Pretty-print guards
498 \B@ifstrequal{\evt@grds}{}{
499   \newcommand\pretty@grds{
500 }{
501   \newcommand\pretty@grds@tmp{
502     \evt@grds~
503   }
504   \B@ifstrequal{\evt@pars}{}{
505     \Bwhen~\pretty@grds@tmp
506   }{
507     \newcommand\pretty@grds{\pretty@grds@tmp}
508   }
509 }
510 % Pretty-print witnesses
511 \B@ifstrequal{\evt@wits}{}{
512   \newcommand\pretty@wits{
513 }{
514   \newcommand\pretty@wits{
515     \Bwith~
516     \evt@wits~
517   }
518 }
519 % Pretty-print actions
520 \B@ifstrequal{\evt@acts}{}{
521   \renewcommand\evt@acts{\SKIP}
522 }{}
523 \newcommand\pretty@acts@tmp{
524   \evt@acts
525 }
526 \newcommand\pretty@acts@keyword{\Bthen}
527 \B@ifstrequal{\evt@pars}{}{
528   \B@ifstrequal{\evt@grds}{}{
529     \renewcommand\pretty@acts@keyword{\Bbegin}
530   }{}
531 }{}
532 \newcommand\pretty@acts{

```

```

533 \pretty@acts@keyword~
534 \pretty@acts@tmp~
535 }
536 % Really do it now
537 \begin{array}{l}
538 \Bevt{\evt@label}\pretty@sts\pretty@absepts~\widehat{=}\sim
539 \pretty@pars
540 \pretty@grds
541 \pretty@wits
542 \pretty@acts
543 \Bend
544 \end{array}
545 } % END group
546 }

547 %%%% INITIALISATION label
548 \newBevt{INITIALISATION}
549
550 %%%% Pretty print the initialisation: no ‘refines’ clause. no parameters, no
551 %%%% guards
552 %%%% Arguments:
553 %%%% 1. (Newline(\))-separated) list of assignments.
554 %%%%
555 %%%% Usage: \initialisation{S1(v,x,y)\S2(w,x,y)}
556 %%%% will produce the following
557 %%%%
558 %%%% init
559 %%%% begin
560 %%%% S1(v, x, y)
561 %%%% S2(w, x, y)
562 %%%% end
563 %%%%
564 \newcommand{\initialisation}[1]{
565 \event{\INITIALISATION}{-}{-}{-}{-}{#1}
566 }
```

# 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	A	\B@fontsize . . . . .
\ 54, 60, 66, 73, 306,	\axioms . . . . . 118	53, 59, 65, 72, 111
330–332, 334,	\axmwdpo . . . . . 189, 190	\B@identifier 80, 96, 274
372, 378, 384,	B	\B@ifstrequal . . . . .
393, 397, 409,	\B@declaration 17, 44,	. 358, 369, 375,
412, 422, 424,	113, 116, 122, 128	381, 387, 395,
427, 436, 553, 555	\B@declarationbase .	405, 416, 425,
	. . . . . 22, 45, 297	426, 480, 486,
	\B@event 11, 41, 334, 359	492, 498, 504,
	\B@eventinline . . . . .	511, 520, 527, 528
\_ . . . . . 62, 68, 75	. . . . . 290, 458, 470	\B@keyword . . 79, 92,

130–138, 299, 306	\Bvrb . . . . . 146, 166	\fboxsep . . . 7, 8, 14,
\B@keywordbase . . . . .	\Bvspace . . 54, 60, 66, 73	18, 19, 25, 29,
. . . . . 79, 93, 273	\Bwhen . . . . 136, 397, 505	30, 36, 286, 287, 293
\B@label . . . . 81, 99, 275	\Bwhere . . . . 137, 384, 495	\finpo . . . . . 230, 231
\B@newmacro 153, 160–171	\Bwith . . . . 138, 409, 515	\fispo . . . . . 222, 223
\B@oldfboxsep . . . . .		\footnotesize . . . . . 65
. . . . . 5, 7, 14, 18,	<b>C</b>	
25, 29, 36, 286, 293	\carriersets . . . . . 112	<b>G</b>
\B@po . . . . . 82, 102, 318	\colorlet . . . 84, 86, 88, 90	\grdpo . . . . 239, 261, 262
\B@section . . . . .	\constants . . . . . 115	\grdthmpo . . . . 211, 212
. . . . . 28, 48, 119, 125	\csname . . . . . 155, 157	
\B@sectionbase . . . . .		<b>H</b>
. . . . . 33, 49, 303	<b>D</b>	\hbox . . . . . 152
\B@tab . . . . . 56,	\DeclareOption . . 39,	\hspace . . . . 55, 61, 67, 74
62, 68, 75, 298,	58, 64, 71, 83, 105	
307, 312, 372,	\def . . . . . 155, 157	<b>I</b>
378, 384, 391,		\ifstrequal . . . . .
397, 409, 410,	<b>E</b>	. . . . . 154, 276, 304, 358
420, 424, 427, 443	\end . . . . . 111, 300, 309,	\INITIALISATION . . . 565
\Bact . . . . . 150, 171, 223, 251	310, 314, 393,	\initialisation 555, 564
\Bany . . . . . 130, 384, 495	412, 422, 444, 544	\invariants . . . . . 124
\Baxm . . . . . 142, 163, 190	\endcsname . . . . 155, 157	\invpo . . . . . 200, 201
\Bbap . . . . . 152	\ensuremath . . . . 10,	
\Bbegin . . . . 131, 427, 529	21, 32, 79–82,	<b>L</b>
\Bcst . . . . . 141, 162	93, 96, 99, 102,	\land . . . . . 454, 455, 458, 461
\Bctx . . . . . 139, 160	139, 145, 289, 318	
\begin . . . . . 111, 298, 305,	\event . . . . . 6, 40, 565	<b>M</b>
307, 312, 391,	\eventB . . . . . 110	\mathbf . . . . 139, 145, 273
410, 420, 435, 537	\eventinline . . . . 285	\mathit . . . . . 274, 279
\Bend . . . . . 132, 443, 543	\evt@absevt 363, 375,	\mathsf . . . . . 277, 318
\Bevt . . . . . 148,	378, 474, 486, 489	
168, 201, 212,	\evt@acts . . . . 367,	<b>N</b>
223, 240, 251,	416, 417, 421,	\natpo . . . . . 270, 271
262, 271, 436, 538	478, 520, 521, 524	\newBact . . . . . 171
\bfseries . . . . . 152	\evt@grds . . . . 365,	\newBaxm . . . . . 163
\Bgrd . . . . . 151, 170, 262	387, 392, 426,	\newBcst . . . . . 162
\Bhspace . . . . 55, 61, 67, 74	476, 498, 502, 528	\newBctx . . . . . 160
\Bidentifier . . 80, 95,	\evt@label . . . . .	\newBevt . . . . . 168, 548
140, 141, 146, 149	. . . . . 362, 436, 473, 538	\newBgrd . . . . . 170
\Bin . . . . . 147, 167, 201	\evt@pars . . . . .	\newBin . . . . . 167
\Blabel . . . . .	. . . . . 364, 381, 384,	\newBmch . . . . . 165
81, 98, 142, 143,	395, 425, 475,	\newBpar . . . . . 169
147, 148, 150, 151	492, 495, 504, 527	\newBset . . . . . 161
\Bmch . . . . . 145, 165	\evt@sts . . . . 361, 369,	\newBthm . . . . . 164
\Bpar . . . . . 149, 169	372, 472, 480, 483	\newBvrb . . . . . 166
\Bpo . . . . . 82, 101,	\evt@wits . . . . 366, 405,	\newcommand 6, 17, 28,
180, 190, 201,	411, 477, 511, 516	53–56, 79–82,
212, 223, 231,	\ExecuteOptions . . .	84, 86, 88, 90,
240, 251, 262, 271	. . . . . 69, 76, 106	110, 112, 115,
\Brefines . . . . 133, 378, 489	\expandafter . . . . .	118, 121, 124,
\Bset . . . . . 140, 161	. . . . . 155, 157, 358	127, 130–143,
\Bstatus . . . . . 134, 372		145–153, 160–
\Bthen . . . . . 135, 424, 526	<b>F</b>	171, 180, 190,
\Bthm . . . . . 143, 164, 180, 212	\fbox . . . . . 9, 20, 31, 288	201, 212, 223,
		231, 240, 251,

262, 271, 273–	\pretty@grds . . . . .	\setBKeywordColour .
275, 285, 297,	. 388, 396, 401,	. . . . . 84, 85
303, 318, 321,	440, 499, 507, 540	\setBLabelColour 88, 89
358, 359, 361–	\pretty@grds@tmp . .	\setBPOColour . . . 90, 91
367, 370, 372,	. . . . 390, 398,	\setlength . 7, 8, 14,
376, 378, 382,	401, 501, 505, 507	18, 19, 25, 29,
384, 388, 390,	\pretty@pars 382, 384,	30, 36, 286, 287, 293
396, 401, 406,	439, 493, 495, 539	\simpo . . . . . 250, 251
408, 419, 424,	\pretty@sts 370, 372,	\SKIP . . . . . 321,
430, 470, 472–	437, 481, 483, 538	357, 417, 469, 521
478, 481, 483,	\pretty@wits 406, 408,	\sl . . . . . 152
487, 489, 493,	441, 512, 514, 541	\small . . . . . 59
495, 499, 501,	\ProcessOptions . . . 109	
507, 512, 514,		<b>T</b>
523, 526, 532, 564	<b>Q</b>	\textcolor 93, 96, 99, 102
\newenvironment . . . 111	\quad . . . . . 56	\textsc . . . . . 321
\newlength . . . . . 5		\thmpo . . . . . 179, 180
\normalsize . . . . . 53	<b>R</b>	<b>V</b>
<b>P</b>	\renewcommand . . 40,	\variables . . . . . 121
\pretty@absepts . . .	44, 48, 59–62,	\variant . . . . . 127
. . . . 376, 378,	65–68, 72–75,	\varpo . . . . . 240
438, 487, 489, 538	92, 95, 98, 101,	
\pretty@acts . . . . .	417, 427, 521, 529	<b>W</b>
. 430, 442, 532, 542	\RequirePackage . . . 1–4	\widehat . . . . . 538
\pretty@acts@keyword		<b>X</b>
. . . . 424, 427,	<b>S</b>	\xspace . . . . . 79–82,
431, 526, 529, 533	\scriptsize . . . . . 72	93, 96, 99, 102,
\pretty@acts@tmp . .	\setBIdentifierColour	110, 139, 145, 318
. 419, 432, 523, 534	. . . . . 86, 87	

## Change History

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