

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`.

```
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]{\[\[#1]} % Vertical space
55 \newcommand{\Bhspace}[1][2em]{\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]{\[\[#1]}
61 \renewcommand{\Bhspace}[1][1em]{\hspace{#1}}
62 \renewcommand{\B@tab}{\ }
63 }
64 \DeclareOption{compact}{
65 \renewcommand{\B@fontsize}{\footnotesize}
66 \renewcommand{\Bvspace}[1][0ex]{\[\[#1]}
67 \renewcommand{\Bhspace}[1][0.5em]{\hspace{#1}}
68 \renewcommand{\B@tab}{\ }
69 \ExecuteOptions{nobox}
70 }
71 \DeclareOption{tiny}{
72 \renewcommand{\B@fontsize}{\scriptsize}
73 \renewcommand{\Bvspace}[1][-0.5ex]{\[\[#1]}
74 \renewcommand{\Bhspace}[1][0.5em]{\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}{ }{
155     %\ifthenelse{equal{#1}{ }}{
156     \expandafter\def\csname #2\endcsname{#3{#2}}
157   }{
158     \expandafter\def\csname #1\endcsname{#3{#2}}
159   }
160 }

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

```

```

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

```

```

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

```

```

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

```



```

317 }
318
319 \newcommand{\B@po}[1]{\ensuremath{\mathsf{\#1}}\xspace}
320
321 %%%% (BEGIN) Macros for Pretty-Print Event-B Components %%%
322 \newcommand{\SKIP}{\textsc{skip}}
323
324
325 %%%% Pretty print an general Event-B event
326 %%%% Arguments:
327 %%%% 1. (Optional) convergence status.
328 %%%% 2. Name of the event.
329 %%%% 3. Name of the abstract event.
330 %%%% 4. (Comma-separated) list of parameters.
331 %%%% 5. (Newline(\))-separated) list of guards.
332 %%%% 6. (Newline(\))-separated) list of witness predicates.
333 %%%% 7. (Newline(\))-separated) list of assignments.
334 %%%%
335 %%%% Usage: \B@event[conv]{conc}{abs}{x,y}{G1(x,y)\G2(x,y)}{W1\W2}{S1(v,x,y)\S2(w,x,y)}
336 %%%% will produce the following
337 %%%%
338 %%%% conc
339 %%%% refines abs
340 %%%% status conv
341 %%%% any x, y where
342 %%%% G1(x, y)
343 %%%% G2(x, y)
344 %%%% with
345 %%%% W1
346 %%%% W2
347 %%%% then
348 %%%% S1(v, x, y)
349 %%%% S2(w, x, y)
350 %%%% end
351 %%%%
352 %%%% Special case:
353 %%%% - Empty abstract event --> refines clause is omitted.
354 %%%% - Empty convergence status --> status clause is omitted.
355 %%%% - Empty witness --> with clause is omitted.
356 %%%% - Empty parameters, empty guards --> begin ... end
357 %%%% - Empty parameters --> when ... then ... end
358 %%%% - Empty actions --> \SKIP
359 \newcommand{\B@event}[7][]{
360   \def\evt@sts{\#1}
361   \def\evt@name{\#2}
362   \def\evt@absevt{\#3}
363   \def\evt@pars{\#4}
364   \def\evt@grds{\#5}
365   \def\evt@wits{\#6}
366   \def\evt@acts{\#7}
367   %% Pretty-print convergence status
368   \ifthenelse{\equal{\evt@sts}{}}{
369     \def\pretty@sts{

```

```

370 }{
371   \def\pretty@sts{\B@tab\Bstatus \B@tab \evt@sts \\\}
372 }
373 % Pretty-print abstract events
374 \ifthenelse{\equal{\evt@absevt}{}}{
375   \def\pretty@absevt{}
376 }{
377   \def\pretty@absevt{\B@tab\Brefines \B@tab \evt@absevt \\\}
378 }
379 % Pretty-print parameters
380 \ifthenelse{\equal{\evt@pars}{}}{
381   \def\pretty@pars{}
382 }{
383   \def\pretty@pars{\B@tab\Bany \B@tab \evt@pars \B@tab \Bwhere \\\}
384 }
385 % Pretty-print guards
386 \ifx\evt@grds\@empty
387   \def\pretty@grds{}
388 \else
389   \def\evt@grds@tmp{
390     \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
391       \evt@grds
392     \end{array}\\\}
393 }
394 \ifthenelse{\equal{\evt@pars}{}}{
395   \def\pretty@grds{
396     \B@tab \Bwhen \\\}
397     \evt@grds@tmp
398   }
399 }{
400   \def\pretty@grds{\evt@grds@tmp}
401 }
402 \fi
403 % Pretty-print witnesses
404 \ifx\evt@wits\@empty
405   \def\pretty@wits{}
406 \else
407   \def\pretty@wits{
408     \B@tab\Bwith\\\}
409     \begin{array}{@{\B@tab\B@tab}l}
410       \evt@wits
411     \end{array}\\\}
412 }
413 \fi
414 % Pretty-print actions
415 \ifx\evt@acts\@empty
416   \def\evt@acts{\SKIP}
417 \else
418   \fi
419   \def\evt@acts@tmp{
420     \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
421       \evt@acts
422     \end{array}\\\}
423 }

```

```

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

```

```

478 %%%% - Empty convergence status --> status clause is omitted.
479 %%%% - Empty witness --> with clause is omitted.
480 %%%% - Empty parameters, empty guards --> begin ... end
481 %%%% - Empty parameters --> when ... then ... end
482 %%%% - Empty actions --> \SKIP
483 \newcommand{\B@eventinline}[7][]{
484   \def\evt@sts{#1}
485   \def\evt@name{#2}
486   \def\evt@absevt{#3}
487   \def\evt@pars{#4}
488   \def\evt@grds{#5}
489   \def\evt@wits{#6}
490   \def\evt@acts{#7}
491   %% Ignore convergence status
492   \def\pretty@sts{
493     % Pretty-print abstract events
494     \ifx\evt@absevt\@empty
495       \def\pretty@absevt{}
496     \else
497       \def\pretty@absevt{\Brefines~\evt@absevt~}
498     \fi
499     % Pretty-print parameters
500     \ifx\evt@pars\@empty
501       \def\pretty@pars{}
502     \else
503       \def\pretty@pars{\Bany~\evt@pars~\Bwhere~}
504     \fi
505     % Pretty-print guards
506     \ifx\evt@grds\@empty
507       \def\pretty@grds{}
508     \else
509       \def\evt@grds@tmp{
510         \evt@grds
511       }
512       \ifx\evt@pars\@empty
513         \def\pretty@grds{
514           \Bwhen~
515           \evt@grds@tmp~
516         }
517       \else
518         \def\pretty@grds{\evt@grds@tmp~}
519       \fi
520     \fi
521     % Pretty-print witnesses
522     \ifx\evt@wits\@empty
523       \def\pretty@wits{}
524     \else
525       \def\pretty@wits{
526         \Bwith~
527         \evt@wits~
528       }
529     \fi
530     % Pretty-print actions
531     \ifx\evt@acts\@empty

```

```

532 \def\evt@acts{\SKIP}
533 \else
534 \fi
535 \def\evt@acts@tmp{
536   \evt@acts
537 }
538 \def\evt@acts@keyword{\Bthen}
539 \ifx\evt@pars\@empty
540 \ifx\evt@grds\@empty
541 \def\evt@acts@keyword{\Bbegin}
542 \else
543 \fi
544 \else
545 \fi
546 \def\pretty@acts{
547   \evt@acts@keyword~
548   \evt@acts@tmp~
549 }
550 % Really do it now
551 \begin{array}{l}
552   \Bevt{\evt@name}\widehat{=}\sim
553   \pretty@sts
554   \pretty@absevs
555   \pretty@pars
556   \pretty@grds
557   \pretty@wits
558   \pretty@acts
559   \Bend
560 \end{array}
561 }
562
563 %%%% INITIALISATION label
564 \newBevt{init}
565
566 %%%% Pretty print the initialisation: no ‘refines’ clause. no parameters, no
567 %%%% guards
568 %%%% Arguments:
569 %%%% 1. (Newline(\))-separated) list of assignments.
570 %%%%
571 %%%% Usage: \init{S1(v,x,y)\S2(w,x,y)}
572 %%%% will produce the following
573 %%%%
574 %%%% init
575 %%%% begin
576 %%%% S1(v, x, y)
577 %%%% S2(w, x, y)
578 %%%% end
579 %%%%
580 \newcommand{\initialisation}[1]{
581   \event{\init}{-}{-}{-}{-}{#1}
582 }

```

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	377, 383, 390,	\csname	156, 158
\@empty 386, 404, 415,	396, 408, 409,		
425, 426, 494,	420, 424, 427, 445	D	
500, 506, 512,	\Bact . 150, 172, 224, 252	\DeclareOption . 39,	
522, 531, 539, 540	\Bany 130, 383, 503	58, 64, 71, 83, 105	
\ 54, 60, 66, 73, 307,	\Baxm 142, 164, 191	\def 156, 158,	
331–333, 335,	\Bbap 152	360–366, 369,	
371, 377, 383,	\Bbegin . . . 131, 427, 541	371, 375, 377,	
392, 396, 408,	\Bcst 141, 163	381, 383, 387,	
411, 422, 424,	\Bctx 139, 161	389, 395, 400,	
427, 438, 455–	\begin 111, 299, 306,	405, 407, 416,	
457, 459, 569, 571	308, 313, 390,	419, 424, 427,	
	409, 420, 437, 551	432, 484–490,	
	\Bend 132, 445, 559	492, 495, 497,	
_ 62, 68, 75	\Bevt 148,	501, 503, 507,	
	169, 202, 213,	509, 513, 518,	
A	224, 241, 252,	523, 525, 532,	
\axioms 118	263, 272, 438, 552	535, 538, 541, 546	
\axmwdpo 190, 191	\bfseries 152		
	\Bgrd 151, 171, 263	E	
B	\Bhspace . 55, 61, 67, 74	\else . 388, 406, 417,	
\B@declaration 17, 44,	\Bidentifier . 80, 95,	428, 430, 496,	
113, 116, 122, 128	140, 141, 146, 149	502, 508, 517,	
\B@declarationbase .	\Binv 147, 168, 202	524, 533, 542, 544	
. 22, 45, 298	\Blabel	\end . . 111, 301, 310,	
\B@event	81, 98, 142, 143,	311, 315, 392,	
11, 41, 335, 359, 459	147, 148, 150, 151	411, 422, 446, 560	
\B@eventinline 291, 483	\Bmch 145, 166	\endcsname . . . 156, 158	
\B@fontsize	\Bpar 149, 170	\ensuremath 10,	
53, 59, 65, 72, 111	\Bpo 82, 101,	21, 32, 79–82,	
\B@identifier 80, 96, 275	181, 191, 202,	93, 96, 99, 102,	
\B@keyword . . 79, 92,	213, 224, 232,	139, 145, 290, 319	
130–138, 300, 307	241, 252, 263, 272	\equal 155, 277, 305,	
\B@keywordbase	\Brefines . 133, 377, 497	368, 374, 380, 394	
. 79, 93, 274	\Bset 140, 162	\event 6, 40, 581	
\B@label . . . 81, 99, 276	\Bstatus 134, 371	\eventB 110	
\B@newmacro 153, 161–172	\Bthen 135, 424, 538	\eventinline 286	
\B@oldfboxsep	\Bthm . 143, 165, 181, 213	\evt@absevt 362, 374,	
. . 5, 7, 14, 18,	\Bvrb 146, 167	377, 486, 494, 497	
25, 29, 36, 287, 294	\Bvspace . 54, 60, 66, 73	\evt@acts 366,	
\B@po 82, 102, 319	\Bwhen 136, 396, 514	415, 416, 421,	
\B@section	\Bwhere . . . 137, 383, 503	490, 531, 532, 536	
. . 28, 48, 119, 125	\Bwith 138, 408, 526	\evt@acts@keyword .	
\B@sectionbase 424, 427,	
. 33, 49, 304	C	433, 538, 541, 547	
\B@tab 56,	\carrier sets 112	\evt@acts@tmp	
62, 68, 75, 299,	\colorlet 84, 86, 88, 90	. 419, 434, 535, 548	
308, 313, 371,	\constants 115		

<code>\evt@grds</code> 364, 386, 391, 426, 488, 506, 510, 540	<code>\initialisation</code> . . . 580	<code>\pretty@sts</code> . . . 369, 371, 439, 492, 553
<code>\evt@grds@tmp</code> 389, 397, 400, 509, 515, 518	<code>\invariants</code> 124	<code>\pretty@wits</code> 405, 407, 443, 523, 525, 557
<code>\evt@name</code> 361, 438, 485, 552	<code>\invpo</code> 201, 202	<code>\ProcessOptions</code> . . . 109
<code>\evt@pars</code> 363, 380, 383, 394, 425, 487, 500, 503, 512, 539	M	Q
<code>\evt@sts</code> 360, 368, 371, 484	<code>\mathbf</code> . . . 139, 145, 274	<code>\quad</code> 56
<code>\evt@wits</code> . 365, 404, 410, 489, 522, 527	<code>\mathit</code> 275, 280	R
<code>\ExecuteOptions</code> 69, 76, 106	<code>\mathsf</code> 278, 319	<code>\renewcommand</code> 40, 44, 48, 59– 62, 65–68, 72– 75, 92, 95, 98, 101
<code>\expandafter</code> . . 156, 158	N	<code>\RequirePackage</code> . . . 1–4
F	<code>\natpo</code> 271, 272	S
<code>\fbox</code> 9, 20, 31, 289	<code>\newBact</code> 172	<code>\scriptsize</code> 72
<code>\fboxsep</code> . . . 7, 8, 14, 18, 19, 25, 29, 30, 36, 287, 288, 294	<code>\newBaxm</code> 164	<code>\setBIdentifierColour</code> 86, 87
<code>\fi</code> . . 402, 413, 418, 429, 431, 498, 504, 519, 520, 529, 534, 543, 545	<code>\newBcst</code> 163	<code>\setBKeywordColour</code> 84, 85
<code>\finpo</code> 231, 232	<code>\newBctx</code> 161	<code>\setBLabelColour</code> 88, 89
<code>\fispo</code> 223, 224	<code>\newBevt</code> 169, 564	<code>\setBPOColour</code> . . . 90, 91
<code>\footnotesize</code> 65	<code>\newBgrd</code> 171	<code>\setlength</code> . 7, 8, 14, 18, 19, 25, 29, 30, 36, 287, 288, 294
G	<code>\newBinv</code> 168	<code>\simpo</code> 251, 252
<code>\grdpo</code> 240, 262, 263	<code>\newBmch</code> 166	<code>\SKIP</code> 322, 358, 416, 482, 532
<code>\grdthmpo</code> 212, 213	<code>\newBpar</code> 170	<code>\sl</code> 152
H	<code>\newBset</code> 162	<code>\small</code> 59
<code>\hbox</code> 152	<code>\newBthm</code> 165	T
<code>\hspace</code> . . 55, 61, 67, 74	<code>\newBvrb</code> 167	<code>\textcolor</code> 93, 96, 99, 102
I	<code>\newcommand</code> 6, 17, 28, 53–56, 79–82, 84, 86, 88, 90, 110, 112, 115, 118, 121, 124, 127, 130– 143, 145–153, 161–172, 181, 191, 202, 213, 224, 232, 241, 252, 263, 272, 274–276, 286, 298, 304, 319, 322, 359, 483, 580	<code>\textsc</code> 322
<code>\ifstrequal</code> 154	<code>\newenvironment</code> . . . 111	<code>\thmpo</code> 180, 181
<code>\ifthenelse</code> 155, 277, 305, 368, 374, 380, 394	<code>\newlength</code> 5	V
<code>\ifx</code> . . 386, 404, 415, 425, 426, 494, 500, 506, 512, 522, 531, 539, 540	<code>\newsize</code> 53	<code>\variables</code> 121
<code>\init</code> 571, 581	P	<code>\variant</code> 127
	<code>\pretty@absepts</code> 375, 377, 440, 495, 497, 554	<code>\varpo</code> 241
	<code>\pretty@acts</code> 432, 444, 546, 558	W
	<code>\pretty@grds</code> . . 387, 395, 400, 442, 507, 513, 518, 556	<code>\widehat</code> 552
	<code>\pretty@pars</code> 381, 383, 441, 501, 503, 555	X
		<code>\xspace</code> 79–82, 93, 96, 99, 102, 110, 139, 145, 319

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