

The `eventB` package*

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

February 21, 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
3	Implementation	1
3.1	Package Loading	1
3.2	Declaration of Options for the Package	2
4	Option for rounding boxes	2

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.

3 Implementation

3.1 Package Loading

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

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

*This document corresponds to `eventB` v1.1, dated 2012/02/21.

3.2 Declaration of Options for the Package

In this part various options for the package are defined.

4 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.

```
3 \newlength{\B@oldfboxsep}
4 \newcommand{\event}[7] [] {
5   \setlength{\B@oldfboxsep}{\fboxsep}
6   \setlength{\fboxsep}{2ex}
7   \fbox{
8     \ensuremath{
9       \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
10    }
11  }
12  \setlength{\fboxsep}{\B@oldfboxsep}
13 }
14
15 \newcommand{\Bdeclaration}[2] {
16   \fbox{
17     \ensuremath{
18       \B@declaration[#1]{#2}
19     }
20   }
21 }
22
23 \newcommand{\Bsection}[3] [] {
24   \setlength{\B@oldfboxsep}{\fboxsep}
25   \setlength{\fboxsep}{2ex}
26   \fbox{
27     \ensuremath{
28       \B@section[#1]{#2}{#3}
29     }
30   }
31   \setlength{\fboxsep}{\B@oldfboxsep}
32 }
33
34 \DeclareOption{nobox}{
35   \renewcommand{\event}[7] [] {
36     \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
37   }
38
39   \renewcommand{\Bdeclaration}[2] {
40     \B@declaration[#1]{#2}
41   }
42
43   \renewcommand{\Bsection}[3] [] {
44     \B@section[#1]{#2}{#3}
45   }
46 }
```

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`.

```

48 \newcommand{\B@fontsize}{\normalsize} % The font size used in Bcode environment
49 \newcommand{\Bvspace}[1][2ex]{\[\[{\#1}] % Vertical space
50 \newcommand{\Bhspace}[1][2em]{\hspace{\#1}} % Horizontal space
51 \newcommand{\Bsep}{\quad} % A small separation space
52
53 \DeclareOption{small}{
54   \renewcommand{\B@fontsize}{\small}
55   \renewcommand{\Bvspace}[1][1ex]{\[\[{\#1}]
56   \renewcommand{\Bhspace}[1][1em]{\hspace{\#1}}
57   \renewcommand{\Bsep}{\ }
58   % ^^A \renewcommand{\eventinline}[7][{}{
59   % ^^A   \B@eventinline{\#1}{\#2}{\#3}{\#4}{\#5}{\#6}{\#7}
60   % ^^A }
61 }
62 \DeclareOption{compact}{
63   \renewcommand{\B@fontsize}{\footnotesize}
64   \renewcommand{\Bvspace}[1][0ex]{\[\[{\#1}]
65   \renewcommand{\Bhspace}[1][0.5em]{\hspace{\#1}}
66   \renewcommand{\Bsep}{\ }
67   \ExecuteOptions{nobox}
68 }
69 \DeclareOption{tiny}{
70   \renewcommand{\B@fontsize}{\scriptsize}
71   \renewcommand{\Bvspace}[1][-0.5ex]{\[\[{\#1}]
72   \renewcommand{\Bhspace}[1][0.5em]{\hspace{\#1}}
73   \renewcommand{\Bsep}{\ }
74   \ExecuteOptions{nobox}
75 }
76

```

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.

```

77 \newcommand{\Bkeyword}[1]{\ensuremath{\B@keyword{\#1}}\xspace}
78 \newcommand{\Bidentifier}[1]{\ensuremath{\B@identifier{\#1}}\xspace}
79 \newcommand{\Blabel}[2][{}]{\ensuremath{\B@label{\#1}{\#2}}\xspace}
80 \newcommand{\Bpo}[1]{\ensuremath{\B@po{\#1}}\xspace}
81 \DeclareOption{colour}{
82   \newcommand{\setBKeywordColour}[1]{\colorlet{B@keywordcolor}{\#1}}
83   \setBKeywordColour{blue}
84   \newcommand{\setBIdentifierColour}[1]{\colorlet{B@identifiercolor}{\#1}}
85   \setBIdentifierColour{blue!50!red}
86   \newcommand{\setBLabelColour}[1]{\colorlet{B@labelcolor}{\#1}}
87   \setBLabelColour{green!50!black}
88   \newcommand{\setBPoColour}[1]{\colorlet{B@pocolor}{\#1}}

```

```

89 \setBP0Colour{red}
90 \renewcommand{\Bkeyword}[1]{
91   \ensuremath{\textcolor{B@keywordcolor}{\B@keyword{#1}}}\xspace
92 }
93 \renewcommand{\Bidentifier}[1]{
94   \ensuremath{\textcolor{B@identifiercolor}{\B@identifier{#1}}}\xspace
95 }
96 \renewcommand{\Blabel}[2] []{
97   \ensuremath{\textcolor{B@labelcolor}{\B@label{#1}{#2}}}\xspace
98 }
99 \renewcommand{\Bpo}[1]{
100   \ensuremath{\textcolor{B@pocolor}{\B@po{#1}}}\xspace
101 }
102 }
103 \DeclareOption{color}{
104   \ExecuteOptions{colour}
105 }
106
107 %%%% Bcode environment %%%%
108 \newenvironment{Bcode}[1] [\B@fontsize]{\begin{center}#1{\end{center}}
109
110
111
112 \newcommand{\B@keyword}[1]{\mathbf{#1}}
113 \newcommand{\B@identifier}[1]{\mathit{#1}}
114 \newcommand{\B@label}[2] []{
115   \def\is@thm{#1}
116   \ifx\is@thm\@empty
117     \mathsf{#2}
118   \else
119     \mathit{#2}
120   \fi
121 }
122
123
124
125 \newcommand{\eventinline}[7] []{
126   \setlength{\B@oldfboxsep}{\fboxsep}
127   \setlength{\fboxsep}{2ex}
128   \fbox{
129     \ensuremath{
130       \B@eventinline{#1}{#2}{#3}{#4}{#5}{#6}{#7}
131     }
132   }
133   \setlength{\fboxsep}{\B@oldfboxsep}
134 }
135
136
137 \newcommand{\B@declaration}[2]{
138   \begin{array}{l@{\Bsep}l}
139     \Bkeyword{#1:} & #2
140   \end{array}
141 }

```

```

142
143 \newcommand{\B@section}[3][{}]{
144   \def\no@title{#1}
145   \ifx\no@title\@empty
146     \begin{array}{l}
147       \Bkeyword{#2:} \\
148       \begin{array}{l@{\Bsep}l}
149         #3
150       \end{array}
151     \end{array}
152   \else
153     \begin{array}{l@{\Bsep}l}
154       #3
155     \end{array}
156   \fi
157 }
158
159 \newcommand{\B@po}[1]{\ensuremath{\mathsf{#1}}\xspace}
160
161 %%%% BEGIN Execution of options %%%%
162 \ProcessOptions
163 %%%% END Execution of options %%%%
164
165 %%%% (BEGIN) Macros for Pretty-Print Event-B Components %%%
166 \newcommand{\eventB}{Event-B\xspace}
167 \newcommand{\SKIP}{\textsc{skip}}
168
169 %%%% Event-B Keywords %%%%
170 \newcommand{\Bany}{\Bkeyword{any}}
171 \newcommand{\Bbegin}{\Bkeyword{begin}}
172 \newcommand{\Bend}{\Bkeyword{end}}
173 \newcommand{\Brefines}{\Bkeyword{refines}}
174 \newcommand{\Bstatus}{\Bkeyword{status}}
175 \newcommand{\Bthen}{\Bkeyword{then}}
176 \newcommand{\Bwhen}{\Bkeyword{when}}
177 \newcommand{\Bwhere}{\Bkeyword{where}}
178 \newcommand{\Bwith}{\Bkeyword{with}}
179
180 %%%% Event-B internal elements %%%%
181 \newcommand{\Bctx}[1]{\ensuremath{\mathbf{#1}}\xspace}
182 \newcommand{\Bset}[1]{\Bidentifier{#1}}
183 \newcommand{\Bcst}[1]{\Bidentifier{#1}}
184 \newcommand{\Baxm}[1]{\Blabel{#1}}
185 \newcommand{\Bthm}[1]{\Blabel{thm}{#1}}
186
187 \newcommand{\Bmch}[1]{\ensuremath{\mathbf{#1}}\xspace}
188 \newcommand{\Bvrb}[1]{\Bidentifier{#1}}
189 \newcommand{\Binv}[1]{\Blabel{#1}}
190 \newcommand{\Bvt}[1]{\Blabel{#1}}
191 \newcommand{\Bpar}[1]{\Bidentifier{#1}}
192 \newcommand{\Bact}[1]{\Blabel{#1}}
193 \newcommand{\Bgrd}[1]{\Blabel{#1}}
194 \newcommand{\Bbap}[1]{\hbox{\sl\bfseries #1}}

```

```

195 %%%
196 %%%% Creating Event-B elements macros %%%%
197
198 %%%% Create a new B macro
199 %%%% Arguments:
200 %%%% 1. The macro string, (OPTIONAL) if empty then the expanded string will be used.
201 %%%% 2. The expanded string
202 %%%% 3. The mark-up macros, e.g. \Bvrb
203 %%%% Usage:
204 %%%% - \B@newmacro[aaa]{a\_a\_a}{\Bvrb} will create a new macro \aaa
205 %%%% which will be expanded to be \Bvrb{a\_a\_a}
206 %%%% - \B@newmacro{aaa}{\Bvrb} will create a new macro \aaa
207 %%%% which will be expanded to be \Bvrb{aaa}
208 %%%%
209 \newcommand{\B@newmacro}[3][{}]{
210   \def\input@macro{#1}
211   \ifx\input@macro\empty
212     \expandafter\def\csname #2\endcsname{#3{#2}}
213   \else
214     \expandafter\def\csname #1\endcsname{#3{#2}}
215   \fi
216 }
217
218 %%%% Create a new context macro
219 %%%% Arguments:
220 %%%% 1. The macro string (OPTIONAL)
221 %%%% 2. The expanded string
222 %%%% Usage:
223 %%%% - \newBctx[aaa]{a\_a\_a} will create a new macro \aaa
224 %%%% which will be expanded to be \Bctx{a\_a\_a}.
225 %%%% - \newBctx{aaa} will create a new macro \aaa which will be
226 %%%% expanded to be \Bctx{aaa}.
227 %%%%
228 \newcommand{\newBctx}[2][{}]{%
229   \B@newmacro[#1]{#2}{\Bctx}
230 }
231
232 %%%% Create a new carrier set macro
233 %%%% Arguments:
234 %%%% 1. The macro string (OPTIONAL)
235 %%%% 2. The expanded string
236 %%%% Usage:
237 %%%% - \newBset[aaa]{a\_a\_a} will create a new macro \aaa
238 %%%% which will be expanded to be \Bset{a\_a\_a}.
239 %%%% - \newBset{aaa} will create a new macro \aaa which will be
240 %%%% expanded to be \Bset{aaa}.
241 %%%%
242 \newcommand{\newBset}[2][{}]{%
243   \B@newmacro[#1]{#2}{\Bset}
244 }
245
246 %%%% Create a new constant macro
247 %%%% Arguments:

```

```

248 %%%% 1. The macro string (OPTIONAL)
249 %%%% 2. The expanded string
250 %%%% Usage:
251 %%%% - \newBcst[aaa]{a\_a\_a} will create a new macro \aaa
252 %%%% which will be expanded to be \Bcst{a\_a\_a}.
253 %%%% - \newBcst{aaa} will create a new macro \aaa which will be
254 %%%% expanded to be \Bcst{aaa}.
255 %%%%
256 \newcommand{\newBcst}[2][]{%
257   \B@newmacro[#1]{#2}{\Bcst}
258 }
259
260 %%%% Create a new axiom macro
261 %%%% Arguments:
262 %%%% 1. The macro string (OPTIONAL)
263 %%%% 2. The expanded string
264 %%%% Usage:
265 %%%% - \newBaxm[aaa]{a\_a\_a} will create a new macro \aaa
266 %%%% which will be expanded to be \Baxm{a\_a\_a}.
267 %%%% - \newBaxm{aaa} will create a new macro \aaa which will be
268 %%%% expanded to be \Baxm{aaa}.
269 %%%%
270 \newcommand{\newBaxm}[2][]{%
271   \B@newmacro[#1]{#2}{\Baxm}
272 }
273
274 %%%% Create a new theorem macro
275 %%%% Arguments:
276 %%%% 1. The macro string (OPTIONAL)
277 %%%% 2. The expanded string
278 %%%% Usage:
279 %%%% - \newBthm[aaa]{a\_a\_a} will create a new macro \aaa
280 %%%% which will be expanded to be \Bthm{a\_a\_a}.
281 %%%% - \newBthm{aaa} will create a new macro \aaa which will be
282 %%%% expanded to be \Bthm{aaa}.
283 %%%%
284 \newcommand{\newBthm}[2][]{%
285   \B@newmacro[#1]{#2}{\Bthm}
286 }
287
288 %%%% Create a new machine macro
289 %%%% Arguments:
290 %%%% 1. The macro string (OPTIONAL)
291 %%%% 2. The expanded string
292 %%%% Usage:
293 %%%% - \newBmch[aaa]{a\_a\_a} will create a new macro \aaa
294 %%%% which will be expanded to be \Bmch{a\_a\_a}.
295 %%%% - \newBmch{aaa} will create a new macro \aaa which will be
296 %%%% expanded to be \Bmch{aaa}.
297 %%%%
298 \newcommand{\newBmch}[2][]{%
299   \B@newmacro[#1]{#2}{\Bmch}
300 }

```

```

301
302 %%%% Create a new variable macro
303 %%%% Arguments:
304 %%%% 1. The macro string (OPTIONAL)
305 %%%% 2. The expanded string
306 %%%% Usage:
307 %%%% - \newBvrb[aaa]{a\_a\_a} will create a new macro \aaa
308 %%%% which will be expanded to be \Bvrb{a\_a\_a}.
309 %%%% - \newBvrb{aaa} will create a new macro \aaa which will be
310 %%%% expanded to be \Bvrb{aaa}.
311 %%%%
312 \newcommand{\newBvrb}[2][]{%
313   \B@newmacro[#1]{#2}{\Bvrb}
314 }
315
316 %%%% Create a new invariant macro
317 %%%% Arguments:
318 %%%% 1. The macro string (OPTIONAL)
319 %%%% 2. The expanded string
320 %%%% Usage:
321 %%%% - \newBinV[aaa]{a\_a\_a} will create a new macro \aaa
322 %%%% which will be expanded to be \BinV{a\_a\_a}.
323 %%%% - \newBinV{aaa} will create a new macro \aaa which will be
324 %%%% expanded to be \BinV{aaa}.
325 %%%%
326 \newcommand{\newBinV}[2][]{%
327   \B@newmacro[#1]{#2}{\BinV}
328 }
329
330 %%%% Create a new event macro
331 %%%% Arguments:
332 %%%% 1. The macro string (OPTIONAL)
333 %%%% 2. The expanded string
334 %%%% Usage:
335 %%%% - \newBevt[aaa]{a\_a\_a} will create a new macro \aaa
336 %%%% which will be expanded to be \Bevt{a\_a\_a}.
337 %%%% - \newBevt{aaa} will create a new macro \aaa which will be
338 %%%% expanded to be \Bevt{aaa}.
339 %%%%
340 \newcommand{\newBevt}[2][]{%
341   \B@newmacro[#1]{#2}{\Bevt}
342 }
343
344 %%%% Create a new parameter macro
345 %%%% Arguments:
346 %%%% 1. The macro string (OPTIONAL)
347 %%%% 2. The expanded string
348 %%%% Usage:
349 %%%% - \newBpar[aaa]{a\_a\_a} will create a new macro \aaa
350 %%%% which will be expanded to be \Bpar{a\_a\_a}.
351 %%%% - \newBpar{aaa} will create a new macro \aaa which will be
352 %%%% expanded to be \Bpar{aaa}.
353 %%%%

```



```

354 \newcommand{\newBpar}[2][]{%
355   \B@newmacro[#1]{#2}{\Bpar}
356 }
357
358 %%%% Create a new guard macro
359 %%%% Arguments:
360 %%%% 1. The macro string (OPTIONAL)
361 %%%% 2. The expanded string
362 %%%% Usage:
363 %%%% - \newBgrd[aaa]{a\_a\_a} will create a new macro \aaa
364 %%%% which will be expanded to be \Bgrd{a\_a\_a}.
365 %%%% - \newBgrd{aaa} will create a new macro \aaa which will be
366 %%%% expanded to be \Bgrd{aaa}.
367 %%%%
368 \newcommand{\newBgrd}[2][]{%
369   \B@newmacro[#1]{#2}{\Bgrd}
370 }
371
372 %%%% Create a new action macro
373 %%%% Arguments:
374 %%%% 1. The macro string (OPTIONAL)
375 %%%% 2. The expanded string
376 %%%% Usage:
377 %%%% - \newBact[aaa]{a\_a\_a} will create a new macro \aaa
378 %%%% which will be expanded to be \Bact{a\_a\_a}.
379 %%%% - \newBact{aaa} will create a new macro \aaa which will be
380 %%%% expanded to be \Bact{aaa}.
381 %%%%
382 \newcommand{\newBact}[2][]{%
383   \B@newmacro[#1]{#2}{\Bact}
384 }
385
386 %%%% Pretty print carrier sets
387 %%%% Arguments:
388 %%%% 1. (Comma-separated) list of carrier sets.
389 %%%%
390 %%%% Usage: \carriersets{S, T}
391 \newcommand{\carriersets}[1]{
392   \Bdeclaration{sets}{#1}
393 }
394
395 %%%% Pretty print constants
396 %%%% Arguments:
397 %%%% 1. (Comma-separated) list of constants.
398 %%%%
399 %%%% Usage: \constants{m, n}
400 \newcommand{\constants}[1]{
401   \Bdeclaration{constants}{#1}
402 }
403
404 %%%% Pretty print axioms
405 %%%% Arguments:
406 %%%% 1. (Newline(\)-separated) list of axioms.

```

```

407 %%%%
408 %%%% Usage: \axioms{\Baxm{axm0\_1}: & x \in \nat \\  

409 %%%% \Baxm{axm0\_2}: & y \in \nat \\[2ex]}
410 \newcommand{\axioms}[2][]{
411 \Bsection[#1]{axioms}{#2}
412 }
413
414 %%%% Pretty print variables
415 %%%% Arguments:
416 %%%% 1. (Comma-separated) list of variables.
417 %%%%
418 %%%% Usage: \variables{x, y}
419 \newcommand{\variables}[1]{
420 \Bdeclaration{variables}{#1}
421 }
422
423 %%%% Pretty print invariants
424 %%%% Arguments:
425 %%%% 1. (Newline(\))-separated) list of invariants.
426 %%%%
427 %%%% Usage: \invariants{\Bin{inv0\_1}: & x \in \nat \\  

428 %%%% \Bin{inv0\_2}: & y \in \nat \\[2ex]}
429 \newcommand{\invariants}[2][]{
430 \Bsection[#1]{invariants}{#2}
431 }
432
433 %%%% Pretty print variant
434 %%%% Arguments:
435 %%%% 1. The variant
436 %%%%
437 %%%% Usage: \variant{V}
438 \newcommand{\variant}[1]{
439 \Bdeclaration{variant}{#1}
440 }
441
442 %%%% Pretty print an general Event-B event
443 %%%% Arguments:
444 %%%% 1. (Optional) convergence status.
445 %%%% 2. Name of the event.
446 %%%% 3. Name of the abstract event.
447 %%%% 4. (Comma-separated) list of parameters.
448 %%%% 5. (Newline(\))-separated) list of guards.
449 %%%% 6. (Newline(\))-separated) list of witness predicates.
450 %%%% 7. (Newline(\))-separated) list of assignments.
451 %%%%
452 %%%% Usage: \B@event[conv]{conc}{abs}{x,y}{G1(x,y)\G2(x,y)}{W1\W2}{S1(v,x,y)\S2(w,x,y)}
453 %%%% will produce the following
454 %%%%
455 %%%% conc
456 %%%% refines abs
457 %%%% status conv
458 %%%% any x, y where
459 %%%% G1(x, y)

```

```

460 %%%% G2(x, y)
461 %%%% with
462 %%%% W1
463 %%%% W2
464 %%%% then
465 %%%% S1(v, x, y)
466 %%%% S2(w, x, y)
467 %%%% end
468 %%%%
469 %%%% Special case:
470 %%%% - Empty abstract event --> refines clause is omitted.
471 %%%% - Empty convergence status --> status clause is omitted.
472 %%%% - Empty witness --> with clause is omitted.
473 %%%% - Empty parameters, empty guards --> begin ... end
474 %%%% - Empty parameters --> when ... then ... end
475 %%%% - Empty actions --> \SKIP
476 \newcommand{\B@event}[7] [] {
477   \def\evt@sts{#1}
478   \def\evt@name{#2}
479   \def\evt@absevt{#3}
480   \def\evt@pars{#4}
481   \def\evt@grds{#5}
482   \def\evt@wits{#6}
483   \def\evt@acts{#7}
484   %% Pretty-print convergence status
485   \ifx\evt@sts\empty
486     \def\pretty@sts{}
487   \else
488     \def\pretty@sts{\Bsep\Bstatus \Bsep \evt@sts \}
489   \fi
490   % Pretty-print abstract events
491   \ifx\evt@absevt\empty
492     \def\pretty@absevt{}
493   \else
494     \def\pretty@absevt{\Bsep\Brefines \Bsep \evt@absevt \}
495   \fi
496   % Pretty-print parameters
497   \ifx\evt@pars\empty
498     \def\pretty@pars{}
499   \else
500     \def\pretty@pars{\Bsep\Bany \Bsep \evt@pars \Bsep \Bwhere \}
501   \fi
502   % Pretty-print guards
503   \ifx\evt@grds\empty
504     \def\pretty@grds{}
505   \else
506     \def\evt@grds@tmp{
507       \begin{array}{@{\Bsep\Bsep}l@{\Bsep}l}
508         \evt@grds
509       \end{array}\}
510   }
511   \ifx\evt@pars\empty
512     \def\pretty@grds{
513       \Bsep \Bwhen \}

```

```

514     \evt@grds@tmp
515 }
516 \else
517 \def\pretty@grds{\evt@grds@tmp}
518 \fi
519 \fi
520 % Pretty-print witnesses
521 \ifx\evt@wits\@empty
522 \def\pretty@wits{}
523 \else
524 \def\pretty@wits{
525     \Bsep\Bwith\\
526     \begin{array}{@{\Bsep\Bsep}ll}
527         \evt@wits
528     \end{array}\\
529 }
530 \fi
531 % Pretty-print actions
532 \ifx\evt@acts\@empty
533 \def\evt@acts{\SKIP}
534 \else
535 \fi
536 \def\evt@acts@tmp{
537     \begin{array}{@{\Bsep\Bsep}l@{\Bsep}l}
538         \evt@acts
539     \end{array}\\
540 }
541 \def\evt@acts@keyword{\Bsep\Bthen \\}
542 \ifx\evt@pars\@empty
543 \ifx\evt@grds\@empty
544 \def\evt@acts@keyword{\Bsep\Bbegin \\}
545 \else
546 \fi
547 \else
548 \fi
549 \def\pretty@acts{
550     \evt@acts@keyword
551     \evt@acts@tmp
552 }
553 % Really do it now
554 \begin{array}{l}
555     \Bevt{\evt@name} \\
556     \pretty@sts
557     \pretty@absevs
558     \pretty@pars
559     \pretty@grds
560     \pretty@wits
561     \pretty@acts
562     \Bsep\Bend
563 \end{array}
564 }
565
566 %%% Pretty print an general Event-B event
567 %%% Arguments:

```

```

568 %%%% 1. (Optional) convergence status.
569 %%%% 2. Name of the event.
570 %%%% 3. Name of the abstract event.
571 %%%% 4. (Comma-separated) list of parameters.
572 %%%% 5. (Newline(\)-separated) list of guards.
573 %%%% 6. (Newline(\)-separated) list of witness predicates.
574 %%%% 7. (Newline(\)-separated) list of assignments.
575 %%%%
576 %%%% Usage: \B@event[conv]{conc}{abs}{x,y}{G1(x,y)\G2(x,y)}{W1\W2}{S1(v,x,y)\S2(w,x,y)}
577 %%%% will produce the following
578 %%%%
579 %%%% conc
580 %%%% refines abs
581 %%%% status conv
582 %%%% any x, y where
583 %%%% G1(x, y)
584 %%%% G2(x, y)
585 %%%% with
586 %%%% W1
587 %%%% W2
588 %%%% then
589 %%%% S1(v, x, y)
590 %%%% S2(w, x, y)
591 %%%% end
592 %%%%
593 %%%% Special case:
594 %%%% - Empty abstract event --> refines clause is omitted.
595 %%%% - Empty convergence status --> status clause is omitted.
596 %%%% - Empty witness --> with clause is omitted.
597 %%%% - Empty parameters, empty guards --> begin ... end
598 %%%% - Empty parameters --> when ... then ... end
599 %%%% - Empty actions --> \SKIP
600 \newcommand{\B@eventinline}[7][]{
601   \def\evt@sts{#1}
602   \def\evt@name{#2}
603   \def\evt@absevt{#3}
604   \def\evt@pars{#4}
605   \def\evt@grds{#5}
606   \def\evt@wits{#6}
607   \def\evt@acts{#7}
608   %% Ignore convergence status
609   \def\pretty@sts{}
610   % Pretty-print abstract events
611   \ifx\evt@absevt\@empty
612     \def\pretty@absevt{}
613   \else
614     \def\pretty@absevt{\Brefines~\evt@absevt~}
615   \fi
616   % Pretty-print parameters
617   \ifx\evt@pars\@empty
618     \def\pretty@pars{}
619   \else
620     \def\pretty@pars{\Bany~\evt@pars~\Bwhere~}
621   \fi

```

```

622 % Pretty-print guards
623 \ifx\evt@grds\@empty
624 \def\pretty@grds{}
625 \else
626 \def\evt@grds@tmp{
627   \evt@grds
628 }
629 \ifx\evt@pars\@empty
630 \def\pretty@grds{
631   \Bwhen~
632   \evt@grds@tmp~
633 }
634 \else
635 \def\pretty@grds{\evt@grds@tmp~}
636 \fi
637 \fi
638 % Pretty-print witnesses
639 \ifx\evt@wits\@empty
640 \def\pretty@wits{}
641 \else
642 \def\pretty@wits{
643   \Bwith~
644   \evt@wits~
645 }
646 \fi
647 % Pretty-print actions
648 \ifx\evt@acts\@empty
649 \def\evt@acts{\SKIP}
650 \else
651 \fi
652 \def\evt@acts@tmp{
653   \evt@acts
654 }
655 \def\evt@acts@keyword{\Bthen}
656 \ifx\evt@pars\@empty
657 \ifx\evt@grds\@empty
658 \def\evt@acts@keyword{\Bbegin}
659 \else
660 \fi
661 \else
662 \fi
663 \def\pretty@acts{
664   \evt@acts@keyword~
665   \evt@acts@tmp~
666 }
667 % Really do it now
668 \begin{array}{l}
669   \Bvt{\evt@name}~\widehat{=}\sim
670   \pretty@sts
671   \pretty@absevs
672   \pretty@pars
673   \pretty@grds
674   \pretty@wits
675   \pretty@acts

```

```

676 \Bend
677 \end{array}
678 }
679
680 %%%% INITIALISATION label
681 \newBevt{init}
682
683 %%%% Pretty print the initialisation: no ‘refines’ clause. no parameters, no
684 %%%% guards
685 %%%% Arguments:
686 %%%% 1. (Newline(\))-separated) list of assignments.
687 %%%%
688 %%%% Usage: \init{S1(v,x,y)\S2(w,x,y)}
689 %%%% will produce the following
690 %%%%
691 %%%% init
692 %%%% begin
693 %%%% S1(v, x, y)
694 %%%% S2(w, x, y)
695 %%%% end
696 %%%%
697 \newcommand{\initialisation}[1]{
698 \event{\init}{-}{-}{-}{\#1}
699 }
700
701 %%%% Theorem Proof Obligation
702 %%%% Print the theorem proof obligation, given the theorem label.
703 %%%% Arguments:
704 %%%% 1. Theorem label
705 %%%%
706 %%%% Usage:
707 %%%% - \thmpo{thm} will produce "thm/THM"
708 \newcommand{\thmpo}[1]{\Bthm{\#1}/\Bpo{THM}}
709
710 %%%% Axiom Well-definedness Proof Obligation
711 %%%% Print the axiom well-definedness proof obligation, given the
712 %%%% axiom label.
713 %%%% Arguments:
714 %%%% 1. Axiom label
715 %%%%
716 %%%% Usage:
717 %%%% - \axmwdpo{axm} will produce "axm/WD"
718 \newcommand{\axmwdpo}[1]{\Baxm{\#1}/\Bpo{WD}}
719
720 %%%% Invariant Proof Obligation
721 %%%% Print the invariant proof obligation, given the event name and
722 %%%% invariant label
723 %%%% Arguments:
724 %%%% 1. Event name
725 %%%% 2. Invariant label
726 %%%%
727 %%%% Usage:
728 %%%% - \invpo{evt}{inv} will produce "evt/inv/INV"
```

```

729 \newcommand{\invpo}[2]{\Bevt{#1}/\Binv{#2}/\Bpo{INV}}
730
731 %%%% Theorem (in guard) Proof Obligation
732 %%%% Print the simulation proof obligation, given the event name and
733 %%%% the theorem (in guard) label.
734 %%%% Arguments:
735 %%%% 1. Event name
736 %%%% 2. Theorem (in guard) label
737 %%%%
738 %%%% Usage:
739 %%%% - \grdthmpo{evt}{thm} will produce "evt/thm/THM"
740 \newcommand{\grdthmpo}[2]{\Bevt{#1}/\Bthm{#2}/\Bpo{THM}}
741
742 %%%% Feasibility Proof Obligation
743 %%%% Print the feasibility proof obligation, given the event name and
744 %%%% the action label
745 %%%% Arguments:
746 %%%% 1. Event name
747 %%%% 2. Action label
748 %%%%
749 %%%% Usage:
750 %%%% - \fispo{evt}{act} will produce "evt/act/FIS"
751 \newcommand{\fispo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{FIS}}
752
753 %%%% Variant finiteness Proof Obligation
754 %%%% Print the Variant finiteness proof obligation
755 %%%% Arguments: No arguments
756 %%%%
757 %%%% Usage:
758 %%%% - \finpo will produce "FIN"
759 \newcommand{\finpo}{\Bpo{FIN}}
760
761 %%%% Variant Proof Obligation
762 %%%% Print the guard strengthen proof obligation, given the event name
763 %%%% Arguments:
764 %%%% 1. Event name
765 %%%%
766 %%%% Usage:
767 %%%% - \grdpo{evt} will produce "evt/VAR"
768 \newcommand{\varpo}[1]{\Bevt{#1}/\Bpo{VAR}}
769
770 %%%% Simulation Proof Obligation
771 %%%% Print the simulation proof obligation, given the event name and
772 %%%% the action label.
773 %%%% Arguments:
774 %%%% 1. Event name
775 %%%% 2. Action label
776 %%%%
777 %%%% Usage:
778 %%%% - \simpo{evt}{act} will produce "evt/act/SIM"
779 \newcommand{\simpo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{SIM}}
780

```



```

781 %%%% Guard Strengthen Proof Obligation
782 %%%% Print the guard strengthen proof obligation, given the event
783 %%%% name and the guard label
784 %%%% Arguments:
785 %%%% 1. (Abstract) Event name
786 %%%% 2. (Abstract) Guard label
787 %%%%
788 %%%% Usage:
789 %%%% - \grdpo{evt}{grd} will produce "evt/grd/GRD"
790 \newcommand{\grdpo}[2]{\Bevt{#1}/\Bgrd{#2}/\Bpo{GRD}}
791
792 %%%% Variant Natural Number Proof Obligation
793 %%%% Print the Variant Natural Number proof obligation, given the event name
794 %%%% Arguments:
795 %%%% 1. Event name
796 %%%%
797 %%%% Usage:
798 %%%% - \natpo{evt} will produce "evt/NAT"
799 \newcommand{\natpo}[1]{\Bevt{#1}/\Bpo{NAT}}
800

```

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	364, 377, 378,	\B@fontsize
\@empty	116,	408, 409, 427, 428	48, 54, 63, 70, 108
145, 211, 485,			\B@identifier 78, 94, 113
491, 497, 503,			\B@keyword . 77, 91, 112
511, 521, 532,	_	57, 66, 73	\B@label . . . 79, 97, 114
542, 543, 611,			\B@newmacro 204, 206,
617, 623, 629,		A	209, 229, 243,
639, 648, 656, 657	\aaa 204, 206, 223, 225,		257, 271, 285,
\ 49, 55, 64, 71, 147,	237, 239, 251,		299, 313, 327,
406, 408, 409,	253, 265, 267,		341, 355, 369, 383
425, 427, 428,	279, 281, 293,		\B@oldfboxsep . 3, 5,
448–450, 452,	295, 307, 309,		12, 24, 31, 126, 133
488, 494, 500,	321, 323, 335,		\B@po 80, 100, 159
509, 513, 525,	337, 349, 351,		\B@section . 28, 44, 143
528, 539, 541,	363, 365, 377, 379		\Bact 192, 378,
544, 555, 572–	\axioms 408, 410		380, 383, 751, 779
574, 576, 686, 688	\axmwdpo 717, 718		\Bany 170, 500, 620
_	204, 205,		\Baxm . 184, 266, 268,
223, 224, 237,		B	271, 408, 409, 718
238, 251, 252,	\B@declaration		\Bbap 194
265, 266, 279, 18, 40, 137		\Bbegin . . . 171, 544, 658
280, 293, 294,	\B@event		\Bcst . 183, 252, 254, 257
307, 308, 321,	9, 36, 452, 476, 576		\Bctx . 181, 224, 226, 229
322, 335, 336,	\B@eventinline		\Bdeclaration 15, 39,
349, 350, 363, 59, 130, 600		392, 401, 420, 439

<code>\begin</code>	108, 138, 146, 148, 153, 507, 526, 537, 554, 668
<code>\Bend</code>	172, 562, 676
<code>\Bevt</code>	190, 336, 338, 341, 555, 669, 729, 740, 751, 768, 779, 790, 799
<code>\bfseries</code>	194
<code>\Bgrd</code>	193, 364, 366, 369, 790
<code>\Bhspace</code>	50, 56, 65, 72
<code>\Bidentifier</code>	78, 93, 182, 183, 188, 191
<code>\Binvs</code>	189, 322, 324, 327, 427, 428, 729
<code>\Bkeyword</code>	77, 90, 139, 147, 170–178
<code>\Blabel</code>	79, 96, 184, 185, 189, 190, 192, 193
<code>\Bmch</code>	187, 294, 296, 299
<code>\Bpar</code>	191, 350, 352, 355
<code>\Bpo</code>	80, 99, 708, 718, 729, 740, 751, 759, 768, 779, 790, 799
<code>\Brefines</code>	173, 494, 614
<code>\Bsection</code>	23, 43, 411, 430
<code>\Bsep</code>	51, 57, 66, 73, 138, 148, 153, 488, 494, 500, 507, 513, 525, 526, 537, 541, 544, 562
<code>\Bset</code>	182, 238, 240, 243
<code>\Bstatus</code>	174, 488
<code>\Bthen</code>	175, 541, 655
<code>\Bthm</code>	185, 280, 282, 285, 708, 740
<code>\Bvrb</code>	188, 202, 204– 207, 308, 310, 313
<code>\Bvspace</code>	49, 55, 64, 71
<code>\Bwhen</code>	176, 513, 631
<code>\Bwhere</code>	177, 500, 620
<code>\Bwith</code>	178, 525, 643
C	
<code>\carriersets</code>	390, 391
<code>\colorlet</code>	82, 84, 86, 88
<code>\constants</code>	399, 400
<code>\csname</code>	212, 214
D	
<code>\DeclareOption</code>	34, 53, 62, 69, 81, 103
<code>\def</code>	115, 144, 210, 212, 214, 477–483, 486, 488, 492, 494, 498, 500, 504, 506, 512, 517, 522, 524, 533, 536, 541, 544, 549, 601–607, 609, 612, 614, 618, 620, 624, 626, 630, 635, 640, 642, 649, 652, 655, 658, 663
E	
<code>\else</code>	118, 152, 213, 487, 493, 499, 505, 516, 523, 534, 545, 547, 613, 619, 625, 634, 641, 650, 659, 661
<code>\end</code>	108, 140, 150, 151, 155, 509, 528, 539, 563, 677
<code>\endcsname</code>	212, 214
<code>\ensuremath</code>	8, 17, 27, 77–80, 91, 94, 97, 100, 129, 159, 181, 187
<code>\event</code>	4, 35, 698
<code>\eventB</code>	166
<code>\eventinline</code>	58, 125
<code>\evt@absepts</code>	479, 491, 494, 603, 611, 614
<code>\evt@acts</code>	483, 532, 533, 538, 607, 648, 649, 653
<code>\evt@acts@keyword</code>	541, 544, 550, 655, 658, 664
<code>\evt@acts@tmp</code>	536, 551, 652, 665
<code>\evt@grds</code>	481, 503, 508, 543, 605, 623, 627, 657
<code>\evt@grds@tmp</code>	506, 514, 517, 626, 632, 635
<code>\evt@name</code>	478, 555, 602, 669
<code>\evt@pars</code>	480, 497, 500, 511, 542, 604, 617, 620, 629, 656
<code>\evt@sts</code>	477, 485, 488, 601
<code>\evt@wits</code>	482, 521, 527, 606, 639, 644
<code>\ExecuteOptions</code>	67, 74, 104
<code>\expandafter</code>	212, 214
F	
<code>\fbox</code>	7, 16, 26, 128
<code>\fboxsep</code>	5, 6, 12, 24, 25, 31, 126, 127, 133
<code>\fi</code>	120, 156, 215, 489, 495, 501, 518, 519, 530, 535, 546, 548, 615, 621, 636, 637, 646, 651, 660, 662
<code>\finpo</code>	758, 759
<code>\fispo</code>	750, 751
<code>\footnotesize</code>	63
G	
<code>\grdpo</code>	767, 789, 790
<code>\grdthmpo</code>	739, 740
H	
<code>\hbox</code>	194
<code>\hspace</code>	50, 56, 65, 72
I	
<code>\ifx</code>	116, 145, 211, 485, 491, 497, 503, 511, 521, 532, 542, 543, 611, 617, 623, 629, 639, 648, 656, 657
<code>\in</code>	408, 409, 427, 428
<code>\init</code>	688, 698
<code>\initialisation</code>	697
<code>\input@macro</code>	210, 211
<code>\invariants</code>	427, 429
<code>\invpo</code>	728, 729
<code>\is@thm</code>	115, 116
M	
<code>\mathbf</code>	112, 181, 187
<code>\mathit</code>	113, 119
<code>\mathsf</code>	117, 159

N		740, 751, 759,	S
\nat ..	408, 409, 427, 428	768, 779, 790, 799	\scriptsize 70
\natpo	798, 799	\newenvironment ... 108	\setBIdentifierColour
\newBact ..	377, 379, 382	\newlength 3 84, 85
\newBaxm ..	265, 267, 270	\no@title 144, 145	\setBKeywordColour .
\newBcst ..	251, 253, 256	\normalsize 48 82, 83
\newBctx ..	223, 225, 228		\setBLabelColour 86, 87
\newBevt		P	\setBPOColour ... 88, 89
. 335, 337, 340, 681		\pretty@absevt ...	\setlength 5, 6, 12, 24,
\newBgrd ..	363, 365, 368 492, 494,	25, 31, 126, 127, 133
\newBinv ..	321, 323, 326	557, 612, 614, 671	\simpo 778, 779
\newBmch ..	293, 295, 298	\pretty@acts	\SKIP 167,
\newBpar ..	349, 351, 354	. 549, 561, 663, 675	475, 533, 599, 649
\newBset ..	237, 239, 242	\pretty@grds .. 504,	\sl 194
\newBthm ..	279, 281, 284	512, 517, 559,	\small 54
\newBvrb ..	307, 309, 312	624, 630, 635, 673	
\newcommand 4, 15, 23,		\pretty@pars 498, 500,	T
48–51, 77–80,		558, 618, 620, 672	\textcolor 91, 94, 97, 100
82, 84, 86, 88,		\pretty@sts ... 486,	\textsc 167
112–114, 125,		488, 556, 609, 670	\thmpo 707, 708
137, 143, 159,		\pretty@wits 522, 524,	
166, 167, 170–		560, 640, 642, 674	V
178, 181–185,		\ProcessOptions ... 162	\variables ... 418, 419
187–194, 209,			\variant 437, 438
228, 242, 256,		Q	\varpo 768
270, 284, 298,		\quad 51	
312, 326, 340,			W
354, 368, 382,		R	\widehat 669
391, 400, 410,		\renewcommand	
419, 429, 438,		35, 39, 43, 54–	X
476, 600, 697,		58, 63–66, 70–	\xspace 77–80,
708, 718, 729,		73, 90, 93, 96, 99	91, 94, 97, 100,
		\RequirePackage .. 1, 2	159, 166, 181, 187

Change History

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