The **algobox** package
Typeset Algobox programs[*]

Julien "_FrnchFrgg_" Rɪᴠᴀᴜᴅ[†]

Released 2019/09/29

1 Documentation

1.1 Getting support

If you find bugs or keywords not recognized by the package, you can report it at the address https://gitlab.com/frnchfrgg-latex/algobox/issues.

1.2 Rationale

French mathematics and/or computer science teachers can use AʟɢᴏBᴏx (http://www.xm1math.net/algobox/) to teach algorithmics to beginners. It uses tree-like indentation, fixed-form structures entered by buttons and dialogs rather than free text. This prevents most typos or small beginner mistakes that are often hard to detect.

AʟɢᴏBᴏx can save a project in its own format, and export in a variety of formats, including LATEX. Several problems arise:

- The output is not similar to what AʟɢᴏBᴏx displays itself.

- The generated code is standalone and uses dependencies that might not be wanted, like algorithm and algpseudocode that can clash with other algorithm-like packages.

- The algorithm part of the generated code does not look like AʟɢᴏBᴏx code and can thus be annoying to modify by hand if needed.

This small package intends to solve the problem by using a syntax very close to the real AʟɢᴏBᴏx syntax and having a typeset output of the best quality possible, resembling the most to what AʟɢᴏBᴏx displays in its editing window.

1.3 Simple usage

To convert an AʟɢᴏBᴏx program to input suitable for this package you just have to:

1. Copy the program from the "Tester l'algorithme" window and remove the line numbers.

---

[*]This file describes v1.3, last revised 2019/09/29.

[†]E-mail: frnchfrgg@free.fr

2. Add `\;` at the end of each line (or at the beginning of each line if like me you prefer to have those out of the way). The reason is that some constructs in AlgoBox programs depend on the line cuts whereas TeX treats those as normal spaces by default. Of course, one can setup TeX so that it temporarily obeys lines, but it *cannot work* if the algorithm is between braces (if you put in in a macro like `\fbox`).

3. Add a backslash at the beginning of all AlgoBox keywords, like `VARIABLES`, `SI`, `DEBUT_SINON` and so on.

4. Remove all underscores in AlgoBox keywords. You probably want to remove them in all cases because AlgoBox programs are not typeset in math mode and TeX will complain that you tried to use a subscript. If you really want an underscore, replace it by `\_` *but not in keywords* or they will not be recognized anymore.

5. Put everything between `\begin{algobox}` and `\end{algobox}`.

Here is an example of the result:

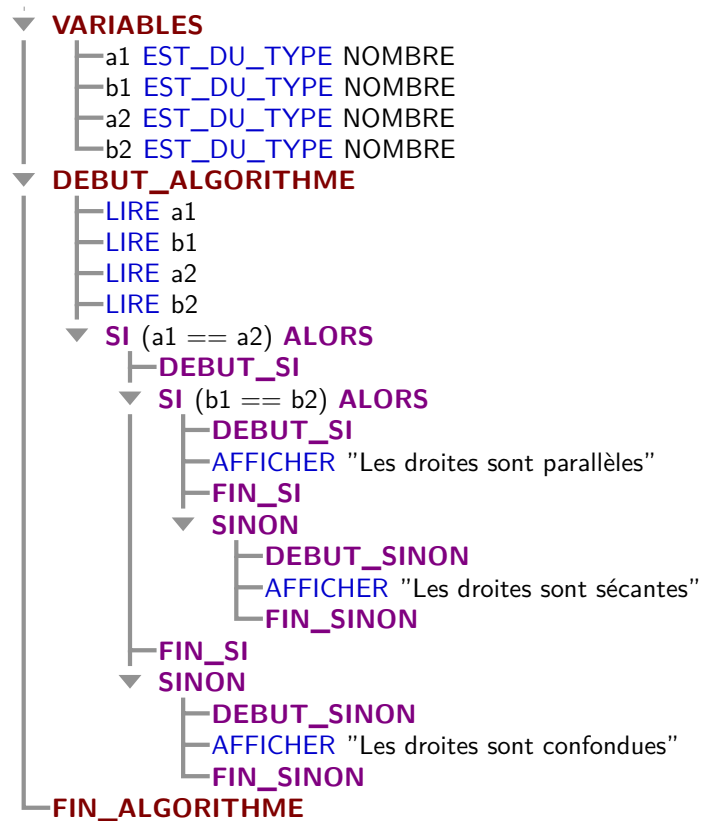```
\begin{algobox}
\;   \VARIABLES
\;   a1 \ESTDUTYPE NOMBRE
\;   b1 \ESTDUTYPE NOMBRE
\;   a2 \ESTDUTYPE NOMBRE
\;   b2 \ESTDUTYPE NOMBRE
\;   \DEBUTALGORITHME
\;   \LIRE a1
\;   \LIRE b1
\;   \LIRE a2
\;   \LIRE b2
\;   \SI (a1 == a2) \ALORS
\;   \DEBUTSI
\;   \SI (b1 == b2) \ALORS
\;   \DEBUTSI
\;   \AFFICHER "Les droites sont parallèles"
\;   \FINSI
\;   \SINON
\;   \DEBUTSINON
\;   \AFFICHER "Les droites sont sécantes"
\;   \FINSINON
\;   \FINSI
\;   \SINON
\;   \DEBUTSINON
\;   \AFFICHER "Les droites sont confondues"
\;   \FINSINON
\; \FINALGORITHME
\end{algobox}
```
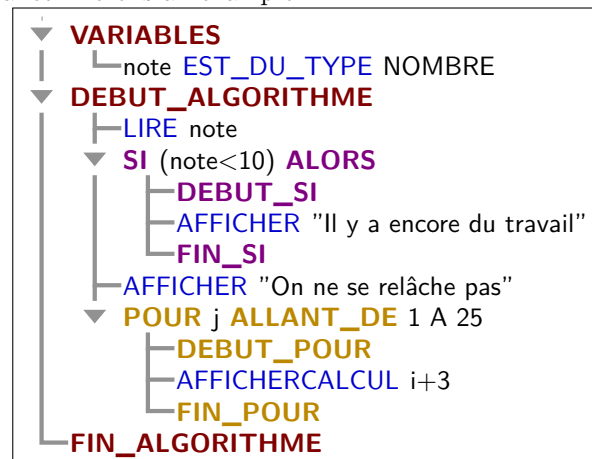
The spacing is only for your convenience as TeX treats any number of consecutive spaces as a single one.

Typesetting the previous example gives:

```
▼ VARIABLES
      ├─a1 EST_DU_TYPE NOMBRE
      ├─b1 EST_DU_TYPE NOMBRE
      ├─a2 EST_DU_TYPE NOMBRE
      └─b2 EST_DU_TYPE NOMBRE
▼ DEBUT_ALGORITHME
      ├─LIRE a1
      ├─LIRE b1
      ├─LIRE a2
      ├─LIRE b2
      ▼ SI (a1 == a2) ALORS
            ├─DEBUT_SI
            ▼ SI (b1 == b2) ALORS
                  ├─DEBUT_SI
                  ├─AFFICHER "Les droites sont parallèles"
                  ├─FIN_SI
                  ▼ SINON
                        ├─DEBUT_SINON
                        ├─AFFICHER "Les droites sont sécantes"
                        └─FIN_SINON
            ├─FIN_SI
            ▼ SINON
                  ├─DEBUT_SINON
                  ├─AFFICHER "Les droites sont confondues"
                  └─FIN_SINON
└─FIN_ALGORITHME
```

The algorithm is typeset as a TikZ picture, so is not breakable between pages.

You can of course put the environment in anything you deem fit, like a \fbox for instance. Here is an example:

```
▼ VARIABLES
 │    └─note EST_DU_TYPE NOMBRE
▼ DEBUT_ALGORITHME
      ├─LIRE note
      ▼ SI (note<10) ALORS
            ├─DEBUT_SI
            ├─AFFICHER "Il y a encore du travail"
            └─FIN_SI
      ├─AFFICHER "On ne se relâche pas"
      ▼ POUR j ALLANT_DE 1 A 25
            ├─DEBUT_POUR
            ├─AFFICHERCALCUL i+3
            └─FIN_POUR
└─FIN_ALGORITHME
```

1.4 Customizing the appearance

TODO

2 **algobox** implementation

```
1  ⟨*package⟩
2  ⟨@@=algobox⟩

3  \ProvidesExplPackage
4    {\ExplFileName}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}

5  \RequirePackage{xparse}
6  \RequirePackage{environ}
7  \RequirePackage{tikz}
8
9  \ExplSyntaxOff
10 \usetikzlibrary{calc}
11 \ExplSyntaxOn
12
13 \RequirePackage{xcolor}
14
15 \int_new:N \l__algobox_level_int
16
```

The colors are defined in `algohighlighter.cpp` in the ALGOBOX source, except for the tree lines color. The names here are the same, with `algobox` prepended.

```
17
18 \definecolor{algoboxarbre}{RGB}{146,146,146}
19 \definecolor{algoboxColorStandard}{HTML}{000000}
20 \definecolor{algoboxColorComment}{HTML}{606060}
21 \definecolor{algoboxColorBloc}{HTML}{800000}
22 \definecolor{algoboxColorCommande}{HTML}{0000CC}
23 \definecolor{algoboxColorSi}{HTML}{800080}
24 \definecolor{algoboxColorTantQue}{HTML}{DD6F06}
25 \definecolor{algoboxColorPour}{HTML}{BB8800}
26 \definecolor{algoboxColorFonction}{HTML}{9A4D00}
27
28 \tikzset{
29     every~algobox/.style=,
30     every~smalgobox/.style={baseline=(block.base)},
31     algobox~ligne/.style={
32         xshift=1em, text~depth=0pt,
33         inner~xsep=0.1ex, inner~ysep=0.5ex
34     },
35     algobox~arbre/.style={ultra~thick, color=algoboxarbre},
36     algobox~indentsymb/.style={x=1ex, y=0.6ex, color=algoboxarbre},
37     algobox~indent/.style={xshift=2em},
38     algobox~normal/.style={
39         font=\sffamily,
40         color=algoboxColorStandard,
41     },
42     algobox~commentaire/.style={
43         algobox~normal/.try,
44         color=algoboxColorComment,
45     },
46     algobox~commande/.style={
47         algobox~normal/.try,
48         color=algoboxColorCommande,
49     },
```

4

```
50    algobox~structure/.style={
51        font=\sffamily\bfseries,
52        color=algoboxColorBloc
53    },
54    algobox~sialors/.style={
55        algobox~structure/.try,
56        color=algoboxColorSi,
57    },
58    algobox~pour/.style={
59        algobox~structure/.try,
60        color=algoboxColorPour,
61    },
62    algobox~tantque/.style={
63        algobox~structure/.try,
64        color=algoboxColorTantQue,
65    },
66    algobox~fonctions/.style={
67        algobox~structure/.try,
68        color=algoboxColorFonction,
69    },
70 }
71
```

2.1 Formatting commands

```
72
73
74 \tl_new:N \l__algobox_lastnode_tl
75 \cs_new_protected:Nn \__algobox_block:nn {
76     \node[algobox~ligne/.try,
77         \tl_if_empty:nF{#1}{algobox~#1/.try},
78         anchor=north~west
79     ]
80         (block) at (A\int_use:N \l__algobox_level_int |- H)
81         {\vphantom{A}#2};
82     \draw[algobox~arbre/.try]
83         (A\int_use:N \l__algobox_level_int)
84         -- (A\int_use:N \l__algobox_level_int |- block.west)
85             coordinate (A\int_use:N \l__algobox_level_int)
86         -- (block.west);
87     \coordinate (H) at (block.south);
88     \tl_set:Nn \l__algobox_lastnode_tl {block}
89 }
90
91 \cs_new_protected:Nn \__algobox_indent:nn {
92     \node[algobox~ligne/.try,
93         \tl_if_empty:nF{#1}{algobox~#1/.try},
94         anchor=north~west
95     ]
96         (block) at (A\int_use:N \l__algobox_level_int |- H)
97         {\vphantom{A}#2};
98     \draw[algobox~arbre/.try]
99         (A\int_use:N \l__algobox_level_int)
100        -- (A\int_use:N \l__algobox_level_int |- H);
101    \fill[algobox~indentsymb/.try]
```

```
102        (A\int_use:N \l__algobox_level_int |- block.west)
103            +(0,-1) -- +(1,1) -- +(-1,1) -- cycle;
104        \coordinate (H) at (block.south);
105        \coordinate (T) at (A\int_use:N \l__algobox_level_int |- H);
106        \coordinate (A\int_use:N \l__algobox_level_int) at (T);
107        \int_incr:N \l__algobox_level_int
108        \coordinate[algobox~indent/.try] (A\int_use:N \l__algobox_level_int) at (T);
109        \tl_set:Nn \l_algobox_lastnode_tl {block}
110  }
111  \cs_new_protected:Nn \__algobox_unindent: {
112        \int_decr:N \l__algobox_level_int
113  }
114
115  \cs_new_protected:Nn \__algobox_node:nn {
116        \node[algobox~#1/.try, anchor=base~west, inner~xsep=0pt]
117            (node) at (\l__algobox_lastnode_tl.base~east) {#2};
118        \tl_set:Nn \l_algobox_lastnode_tl {node}
119  }
120
```

2.2 Parsing helpers

```
121
122
123  \cs_new_protected:Nn \__algobox_parseall:n {
124        \tl_set:Nn \l_tmpa_tl {#1}
125        \seq_set_split:NnV \l_tmpa_seq {\;} \l_tmpa_tl
126        % Now build up the algorithm body, working around the few infix notations.
127        \tl_clear:N \l__algobox_body_tl
128        \seq_map_variable:NNn \l_tmpa_seq \l_tmpa_tl {
129            \tl_if_in:NnT \l_tmpa_tl {\ESTDUTYPE} {
130                \tl_put_left:Nn \l_tmpa_tl {\__algobox_ESTDUTYPE:w}
131            }
132            \tl_if_in:NnT \l_tmpa_tl {\PRENDLAVALEUR} {
133                \tl_put_left:Nn \l_tmpa_tl {\__algobox_PRENDLAVALEUR:w}
134            }
135            \tl_put_right:Nn \l_tmpa_tl { \; }
136            \tl_put_right:NV \l__algobox_body_tl \l_tmpa_tl
137        }
138        \tl_use:N \l__algobox_body_tl
139  }
140
141  \bool_new:N \l__algobox_balanced_first_bool
142  \cs_new_protected:Npn \__algobox_new_balanced_command:nNNnn #1#2#3 {
143        \cs_new_protected:cpn {#1} ##1 #2 {
144            \tl_set:Nn \l__algobox_prefix_tl { ##1 }
145            \tl_clear:N \l__algobox_balanced_tl
146            \bool_set_true:N \l__algobox_balanced_first_bool
147            \use:c {#1_stage2} #2
148        }
149        \cs_new_protected:cpn {#1_stage2} ##1 #2 ##2 #3 {
150            \tl_put_right:Nn \l__algobox_balanced_tl { ##1 }
151            \bool_if:NF \l__algobox_balanced_first_bool {
152                \tl_put_right:Nn \l__algobox_balanced_tl { #2 }
153            }
```

```
154        \tl_if_in:nnTF {##2} {#2} {
155            \bool_set_false:N \l__algobox_balanced_first_bool
156            \use:c {#1_stage2} ##2 \q_mark
157        }{
158            \tl_put_right:Nn \l__algobox_balanced_tl { ##2 }
159            \tl_replace_all:Nnn \l__algobox_balanced_tl {\q_mark} {#3}
160            \tl_set:Nx \l_tmpa_tl {
161                \exp_not:c {#1_stage3}
162                \exp_not:V \l__algobox_prefix_tl
163                \exp_not:n {#2}
164                {
165                    \exp_not:V \l__algobox_balanced_tl
166                }
167            }
168            \tl_use:N \l_tmpa_tl
169        }
170    }
171    \exp_args:Nc \NewDocumentCommand {#1_stage3}
172 }
173
174 \cs_new_protected:Npn \__algobox_if_next:NTF #1#2#3 {
175    \peek_meaning_remove_ignore_spaces:NTF \; {
176        \__algobox_if_next:NTF #1 {#2} {#3}
177    }{
178        \peek_meaning_remove_ignore_spaces:NTF \par {
179            \__algobox_if_next:NTF #1 {#2} {#3}
180        }{
181            \peek_meaning_ignore_spaces:NTF #1 {#2} {#3}
182        }
183    }
184 }
```

2.3 Parsing/typesetting commands

```
185
186 \NewDocumentCommand \__algobox_ESTDUTYPE:w { +u{\ESTDUTYPE} +u{\;} } {
187    \__algobox_block:nn {normal} {
188        \ignorespaces #1 \unskip\c_space_token\null
189    }
190    \__algobox_node:nn {commande}{EST_DU_TYPE}
191    \__algobox_node:nn {normal}{\null\c_space_token#2}
192 }
193
194 \NewDocumentCommand \__algobox_FINALGORITHME:w {} {
195    \int_while_do:nNnn \l__algobox_level_int > \c_zero_int {
196        \__algobox_unindent:
197    }
198    \__algobox_block:nn {structure} {FIN_ALGORITHME}
199 }
200
201 \NewDocumentCommand \__algobox_PRENDLAVALEUR:w { u{\PRENDLAVALEUR} m } {
202    \__algobox_block:nn {normal} {
203        \ignorespaces#1\unskip\c_space_token\null
204    }
205    \__algobox_node:nn {commande}{PREND_LA_VALEUR}
```

```
206     \__algobox_node:nn {normal}{\null\c_space_token#2}
207 }
208
209 \__algobox_new_balanced_command:nNNnn
210         {__algobox_SI:w} \DEBUTSI \FINSI { r() u{\DEBUTSI} +m }
211 {
212     \__algobox_indent:nn {sialors} {SI}
213     \__algobox_node:nn {normal}{\null\c_space_token(#1)\c_space_token\null}
214     \__algobox_node:nn {sialors}{ALORS}
215     \__algobox_block:nn {sialors} {DEBUT_SI}
216     #3
217     \__algobox_block:nn {sialors} {FIN_SI}
218     \__algobox_if_next:NTF \SINON {
219         \__algobox_SINON:w
220     }{
221         \__algobox_unindent:
222     }
223 }
224 \__algobox_new_balanced_command:nNNnn
225         {__algobox_SINON:w} \DEBUTSINON \FINSINON { u{\DEBUTSINON} +m } {
226     \__algobox_indent:nn {sialors} {SINON}
227     \__algobox_block:nn {sialors} {DEBUT_SINON}
228     #2
229     \__algobox_block:nn {sialors} {FIN_SINON}
230     \__algobox_unindent:
231     \__algobox_unindent:
232 }
233
234 \__algobox_new_balanced_command:nNNnn
235         {__algobox_POUR:w} \DEBUTPOUR \FINPOUR
236         { u{\ALLANTDE} u{\A} u{\DEBUTPOUR} +m } {
237     \__algobox_indent:nn {pour} {POUR}
238     \__algobox_node:nn {normal}{\null\c_space_token #1\unskip\c_space_token\null}
239     \__algobox_node:nn {pour}{ALLANT_DE}
240     \__algobox_node:nn {normal}{\null\c_space_token #2\unskip
241                     \c_space_token A~#3\unskip\c_space_token\null}
242     \__algobox_block:nn {pour} {DEBUT_POUR}
243     #4
244     \__algobox_block:nn {pour} {FIN_POUR}
245     \__algobox_unindent:
246 }
247
248 \__algobox_new_balanced_command:nNNnn {__algobox_TANTQUE:w} \DEBUTTANTQUE \FINTANTQUE
249         { r() u{\DEBUTTANTQUE} +m }
250 {
251     \__algobox_indent:nn {tantque} {TANT_QUE}
252     \__algobox_node:nn {normal}{\null\c_space_token(#1)\c_space_token\null}
253     \__algobox_node:nn {tantque}{FAIRE}
254     \__algobox_block:nn {tantque} {DEBUT_TANT_QUE}
255     #3
256     \__algobox_block:nn {tantque} {FIN_TANT_QUE}
257     \__algobox_unindent:
258 }
259
```

```
260 \NewDocumentCommand \__algobox_FONCTION:w { +u{\;} } {
261     \int_while_do:nNnn \l__algobox_level_int > \c_one_int {
262         \__algobox_unindent:
263     }
264     \__algobox_indent:nn {fonctions} {FONCTION}
265     \__algobox_node:nn {normal}{\null\c_space_token#1}
266 }
267
268 \NewDocumentCommand \__algobox_LINE:w { s O{normal} m } {
269     \IfBooleanTF {#1} {
270         \__algobox_indent:nn {#2} {#3}
271         \__algobox_unindent:
272     }{
273         \__algobox_block:nn {#2} {#3}
274     }
275 }
276
277 \NewDocumentCommand \__algobox_NODE:w { O{normal} m } {
278     \__algobox_node:nn {#1} {#2}
279 }
280
```

2.4   User commands and environment

```
281
282 \cs_new_protected_nopar:Nn \__algobox_make_keyword:n {
283     \tl_set:Nn \l_tmpa_tl {#1}
284     \tl_replace_all:Nnn \l_tmpa_tl {_} {}
285     \cs_set:cpn {\l_tmpa_tl} {\__algobox_keyword:nw {#1}}
286 }
287
288 \NewDocumentCommand \__algobox_keyword:nw {m +u{\;}} {
289     \__algobox_block:nn {commande} {#1}
290     \__algobox_node:nn {normal}{\null\c_space_token#2}
291 }
292
293 \cs_new_protected_nopar:Nn \__algobox_make_structure:nn {
294     \tl_set:Nn \l_tmpa_tl {#2}
295     \tl_replace_all:Nnn \l_tmpa_tl {_} {}
296     \cs_set:cpn {\l_tmpa_tl} {\__algobox_topstructure:nn {#1}{#2}}
297 }
298
299 \cs_new_protected_nopar:Nn \__algobox_topstructure:nn {
300     \int_while_do:nNnn \l__algobox_level_int > \c_zero_int {
301         \__algobox_unindent:
302     }
303     \__algobox_indent:nn {#1} {#2}
304 }
305
306 \cs_new_protected_nopar:Nn \__algobox_make_func_structure:nn {
307     \tl_set:Nn \l_tmpa_tl {#1}
308     \tl_replace_all:Nnn \l_tmpa_tl {_} {}
309     \cs_set:cpn {\l_tmpa_tl} {\__algobox_funcstructure:nn {#1}{#2}}
310 }
311
```

```
312 \cs_new_protected_nopar:Nn \__algobox_funcstructure:nn {
313     \int_while_do:nNnn \l__algobox_level_int > { 2 } {
314         \__algobox_unindent:
315     }
316     \bool_if:nTF {#2}{\__algobox_indent:nn}{\__algobox_block:nn} {fonctions} {#1}
317 }
318
319 \clist_const:Nn \c__algobox_keywords_clist {
320     PAUSE, LIRE, AFFICHERCALCUL, AFFICHER,
321     TRACER_POINT, TRACER_POINT_Rouge, TRACER_POINT_Vert,
322     TRACER_POINT_Bleu, TRACER_POINT_Blanc,
323     TRACER_SEGMENT, TRACER_SEGMENT_Rouge, TRACER_SEGMENT_Vert,
324     TRACER_SEGMENT_Bleu, TRACER_SEGMENT_Blanc,
325     RENVOYER, APPELER_FONCTION
326 }
327
328 \cs_new_protected_nopar:Nn \__algobox_begincode:n {
329     \int_zero:N \l__algobox_level_int
330     \begin{tikzpicture}[#1]
331         \coordinate (H) at (0,0);
332         \coordinate (A\int_use:N \l__algobox_level_int) at (0,0);
333         \cs_set_eq:NN \par \prg_do_nothing:
334         \cs_set_eq:NN \; \prg_do_nothing:
335         \cs_set_eq:NN \LINE            \__algobox_LINE:w
336         \cs_set_eq:NN \NODE            \__algobox_NODE:w
337         \cs_set_eq:NN \FINALGORITHME   \__algobox_FINALGORITHME:w
338         \cs_set_eq:NN \SI              \__algobox_SI:w
339         \cs_set_eq:NN \POUR            \__algobox_POUR:w
340         \cs_set_eq:NN \TANTQUE         \__algobox_TANTQUE:w
341         \cs_set_eq:NN \FONCTION        \__algobox_FONCTION:w
342         \cs_set:Npn \//##1\;           {\__algobox_block:nn{commentaire}{//##1}}
343         \clist_map_function:NN \c__algobox_keywords_clist \__algobox_make_keyword:n
344         \__algobox_make_structure:nn {structure} {VARIABLES}
345         \__algobox_make_structure:nn {structure} {DEBUT_ALGORITHME}
346         \__algobox_make_structure:nn {fonctions} {FONCTIONS_UTILISEES}
347         \__algobox_make_func_structure:nn {VARIABLES_FONCTION} \c_true_bool
348         \__algobox_make_func_structure:nn {DEBUT_FONCTION} \c_false_bool
349         \__algobox_make_func_structure:nn {FIN_FONCTION} \c_false_bool
350 }
351 \cs_new_protected_nopar:Nn \__algobox_closecode: {
352     \end{tikzpicture}
353 }
354
355
356
357 \NewDocumentCommand \smalgobox { O{} m } {
358     \group_begin:
359     \__algobox_begincode:n {every~smalgobox/.try,#1}
360     \__algobox_parseall:n {#2}
361     \__algobox_closecode:
362     \group_end:
363 }
364
365 \NewDocumentEnvironment {algobox} {O{}} {
```

```
366    \cs_set_protected:Nn \__algobox_algobox:n {
367        \__algobox_begincode:n {every~algobox/.try,#1}
368        \__algobox_parseall:n {##1}
369        \__algobox_closecode:
370    }
371    \Collect@Body\__algobox_algobox:n
372 }{
373 }
374 ⟨/package⟩
```