

The **algotbox** package

Typeset Algotbox programs*

Julien “_FrncHfrgg_” RIVAUD†

Released 2017/09/03

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/algotbox/issues>.

1.2 Rationale

French mathematics and/or computer science teachers can use ALGOTBOX (<http://www.xm1math.net/algotbox/>) 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.

ALGOTBOX can save a project in its own format, and export in a variety of formats, including L^AT_EX. Several problems arise:

- The output is not similar to what ALGOTBOX 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 ALGOTBOX 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 ALGOTBOX syntax and having a typeset output of the best quality possible, resembling the most to what ALGOTBOX displays in its editing window.

1.3 Simple usage

To convert an ALGOTBOX 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 v1.1, last revised 2017/09/03.

†E-mail: frnchfrgg@free.fr

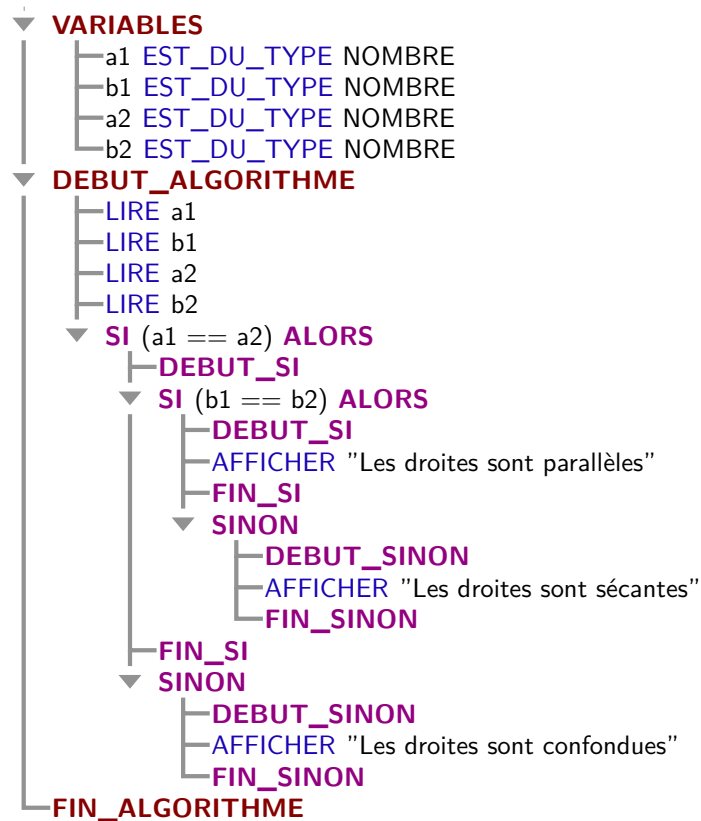
2. Add \; at the end of each line (or at the beginning of each line if like me you prefer to have those out of the way). The reason is that some constructs in ALGOBOX programs depend on the line cuts whereas T_EX treats those as normal spaces by default. Of course, one can setup T_EX so that it temporarily 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 T_EX 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{algorithme} and \end{algorithme}.

Here is an example of the result:

```
\begin{algorithme}
\;   \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{algorithme}
```

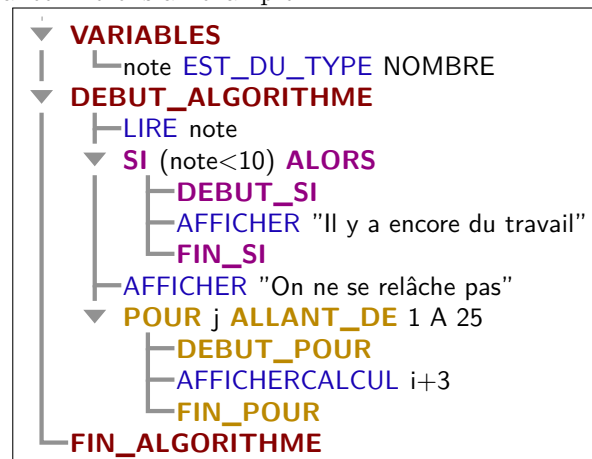
The spacing is only for your convenience as T_EX treats any number of consecutive spaces as a single one.

Typesetting the previous example gives:



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:



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
17 \definecolor{algoboxarbre}{RGB}{146,146,146}
18 \colorlet{algoboxcommentaire}{algoboxarbre!70!black}
19 \definecolor{algoboxstructure}{RGB}{134,0,0}
20 \definecolor{algoboxmotclef}{RGB}{25,3,180}
21 \definecolor{algoboxsialors}{RGB}{147,0,128}
22 \definecolor{algoboxpour}{RGB}{187,136,0}
23 \definecolor{algoboxtantque}{RGB}{222,111,6}
24 \definecolor{algoboxnormal}{RGB}{0,0,0}
25
26 \tikzset{
27   every~algobox/.style=,
28   every~smalgobox/.style={baseline=(block.base)},
29   algobox~ligne/.style={
30     xshift=1em, text-depth=0pt,
31     inner~xsep=0.1ex, inner~ysep=0.5ex
32   },
33   algobox~arbre/.style={ultra~thick, color=algoboxarbre},
34   algobox~indentsymb/.style={x=1ex, y=0.6ex, color=algoboxarbre},
35   algobox~indent/.style={xshift=2em},
36   algobox~normal/.style={
37     font=\sffamily,
38     color=algoboxnormal,
39   },
40   algobox~commentaire/.style={
41     algobox~normal/.try,
42     color=algoboxcommentaire,
43   },
44   algobox~motclef/.style={
45     algobox~normal/.try,
46     color=algoboxmotclef,
47   },
48   algobox~structure/.style={
49     font=\sffamily\bfseries,
50     color=algoboxstructure
51   },
52   algobox~sialors/.style={

```

```

53         algobox-structure/.try,
54         color=algoboxsialors,
55     },
56     algobox-pour/.style={
57         algobox-structure/.try,
58         color=algoboxpour,
59     },
60     algobox-tantque/.style={
61         algobox-structure/.try,
62         color=algoboxtantque,
63     },
64 }
65

```

2.1 Formatting commands

```

66
67
68 \tl_new:N \l__algobox_lastnode_tl
69 \cs_new_protected:Nn \__algobox_block:nn {
70     \node[algobox~ligne/.try,
71         \tl_if_empty:nF{#1}{algobox~#1/.try},
72         anchor=north-west
73     ]
74         (block) at (A\int_use:N \l__algobox_level_int |- H)
75         {\vphantom{A}#2};
76     \draw[algobox~arbre/.try]
77         (A\int_use:N \l__algobox_level_int)
78         -- (A\int_use:N \l__algobox_level_int |- block.west)
79         coordinate (A\int_use:N \l__algobox_level_int)
80         -- (block.west);
81     \coordinate (H) at (block.south);
82     \tl_set:Nn \l__algobox_lastnode_tl {block}
83 }
84
85 \cs_new_protected:Nn \__algobox_indent:nn {
86     \node[algobox~ligne/.try,
87         \tl_if_empty:nF{#1}{algobox~#1/.try},
88         anchor=north-west
89     ]
90         (block) at (A\int_use:N \l__algobox_level_int |- H)
91         {\vphantom{A}#2};
92     \draw[algobox~arbre/.try]
93         (A\int_use:N \l__algobox_level_int)
94         -- (A\int_use:N \l__algobox_level_int |- H);
95     \fill[algobox~indentsymb/.try]
96         (A\int_use:N \l__algobox_level_int |- block.west)
97         +(0,-1) -- +(1,1) -- +(-1,1) -- cycle;
98     \coordinate (H) at (block.south);
99     \coordinate (T) at (A\int_use:N \l__algobox_level_int |- H);
100     \coordinate (A\int_use:N \l__algobox_level_int) at (T);
101     \int_incr:N \l__algobox_level_int
102     \coordinate[algobox~indent/.try] (A\int_use:N \l__algobox_level_int) at (T);
103     \tl_set:Nn \l__algobox_lastnode_tl {block}
104 }

```

```

105 \cs_new_protected:Nn \__algobox_unindent: {
106     \int_decr:N \l__algobox_level_int
107 }
108
109 \cs_new_protected:Nn \__algobox_node:nn {
110     \node[algobox~#1/.try, anchor=base-west, inner~xsep=0pt]
111         (node) at (\l__algobox_lastnode_tl.base-east) {#2};
112     \tl_set:Nn \l__algobox_lastnode_tl {node}
113 }
114

```

2.2 Parsing helpers

```

115
116
117 \cs_new_protected:Nn \__algobox_parseall:n {
118     \tl_set:Nn \l_tmpa_tl {#1}
119     \seq_set_split:NnV \l_tmpa_seq {\;} \l_tmpa_tl
120     % Now build up the algorithm body, working around the few infix notations.
121     \tl_clear:N \l__algobox_body_tl
122     \seq_map_variable:NNn \l_tmpa_seq \l_tmpa_tl {
123         \tl_if_in:NnT \l_tmpa_tl {\ESTDUTYPE} {
124             \tl_put_left:Nn \l_tmpa_tl {\__algobox_ESTDUTYPE:w}
125         }
126         \tl_if_in:NnT \l_tmpa_tl {\PRENDLAVALEUR} {
127             \tl_put_left:Nn \l_tmpa_tl {\__algobox_PRENDLAVALEUR:w}
128         }
129         \tl_put_right:Nn \l_tmpa_tl { \; }
130         \tl_put_right:NV \l__algobox_body_tl \l_tmpa_tl
131     }
132     \tl_use:N \l__algobox_body_tl
133 }
134
135 \bool_new:N \l__algobox_balanced_first_bool
136 \cs_new_protected:Npn \__algobox_new_balanced_command:nNNnn #1#2#3 {
137     \cs_new_protected:cpn {#1} ##1 #2 {
138         \tl_set:Nn \l__algobox_prefix_tl { ##1 }
139         \tl_clear:N \l__algobox_balanced_tl
140         \bool_set_true:N \l__algobox_balanced_first_bool
141         \use:c {#1_stage2} #2
142     }
143     \cs_new_protected:cpn {#1_stage2} ##1 #2 ##2 #3 {
144         \tl_put_right:Nn \l__algobox_balanced_tl { ##1 }
145         \bool_if:NF \l__algobox_balanced_first_bool {
146             \tl_put_right:Nn \l__algobox_balanced_tl { #2 }
147         }
148         \tl_if_in:nnTF {##2} {#2} {
149             \bool_set_false:N \l__algobox_balanced_first_bool
150             \use:c {#1_stage2} ##2 \q_mark
151         }{
152             \tl_put_right:Nn \l__algobox_balanced_tl { ##2 }
153             \tl_replace_all:Nnn \l__algobox_balanced_tl {\q_mark} {#3}
154             \tl_set:Nx \l_tmpa_tl {
155                 \exp_not:c {#1_stage3}
156                 \exp_not:V \l__algobox_prefix_tl

```

```

157             \exp_not:n {#2}
158             {
159                 \exp_not:V \l__algorithmbalanced_tl
160             }
161         }
162         \tl_use:N \l_tmpa_tl
163     }
164 }
165 \exp_args:Nc \NewDocumentCommand {#1_stage3}
166 }
167
168 \cs_new_protected:Npn \__algorithmbalanced_if_next:NTF #1#2#3 {
169     \peek_meaning_remove_ignore_spaces:NTF \; {
170         \__algorithmbalanced_if_next:NTF #1 {#2} {#3}
171     }{
172         \peek_meaning_remove_ignore_spaces:NTF \par {
173             \__algorithmbalanced_if_next:NTF #1 {#2} {#3}
174         }{
175             \peek_meaning_ignore_spaces:NTF #1 {#2} {#3}
176         }
177     }
178 }

```

2.3 Parsing/typesetting commands

```

179
180 \NewDocumentCommand \__algorithmbalanced_VARIABLES:w {} {
181     \__algorithmbalanced_indent:nn {structure} {VARIABLES}
182 }
183
184 \NewDocumentCommand \__algorithmbalanced_ESTDUTYPE:w { +u{\ESTDUTYPE} +u{\;} } {
185     \__algorithmbalanced_block:nn {normal} {
186         \ignorespaces #1 \unskip\c_space_token\newline
187     }
188     \__algorithmbalanced_node:nn {motclef}{EST_DU_TYPE}
189     \__algorithmbalanced_node:nn {normal}{\null\c_space_token#2}
190 }
191
192 \NewDocumentCommand \__algorithmbalanced_DEBUTALGORITHME:w {} {
193     \int_compare:nNnT \l__algorithmbalanced_level_int = \c_one {
194         % if there was a VARIABLES before
195         \__algorithmbalanced_unindent:
196     }
197     \__algorithmbalanced_indent:nn {structure} {DEBUT_ALGORITHME}
198 }
199
200 \NewDocumentCommand \__algorithmbalanced_FINALALGORITHME:w {} {
201     \int_compare:nNnT \l__algorithmbalanced_level_int = \c_one {
202         \__algorithmbalanced_unindent:
203         \__algorithmbalanced_block:nn {structure} {FIN_ALGORITHME}
204     }
205 }
206
207 \NewDocumentCommand \__algorithmbalanced_PRENDLAVALEUR:w { u{\PRENDLAVALEUR} m } {
208     \__algorithmbalanced_block:nn {normal} {

```

```

209         \ignorespaces#1\unskip\c_space_token\null
210     }
211     \__algorithme_node:nn {motclef}{PREND_LA_VALEUR}
212     \__algorithme_node:nn {normal}{\null\c_space_token#2}
213 }
214
215 \__algorithme_new_balanced_command:nNNnn
216     {__algorithme_SI:w} \DEBUTSI \FINSI { r() u{\DEBUTSI} +m }
217 {
218     \__algorithme_indent:nn {sialors} {SI}
219     \__algorithme_node:nn {normal}{\null\c_space_token(#1)\c_space_token\null}
220     \__algorithme_node:nn {sialors}{ALORS}
221     \__algorithme_block:nn {sialors} {DEBUT_SI}
222     #3
223     \__algorithme_block:nn {sialors} {FIN_SI}
224     \__algorithme_if_next:NTF \SINON {
225         \__algorithme_SINON:w
226     }{
227         \__algorithme_unindent:
228     }
229 }
230 \__algorithme_new_balanced_command:nNNnn
231     {__algorithme_SINON:w} \DEBUTSINON \FINSINON { u{\DEBUTSINON} +m } {
232     \__algorithme_indent:nn {sialors} {SINON}
233     \__algorithme_block:nn {sialors} {DEBUT_SINON}
234     #2
235     \__algorithme_block:nn {sialors} {FIN_SINON}
236     \__algorithme_unindent:
237     \__algorithme_unindent:
238 }
239
240 \__algorithme_new_balanced_command:nNNnn
241     {__algorithme_POUR:w} \DEBUTPOUR \FINPOUR
242     { u{\ALLANTDE} u{\A} u{\DEBUTPOUR} +m } {
243     \__algorithme_indent:nn {pour} {POUR}
244     \__algorithme_node:nn {normal}{\null\c_space_token #1\unskip\c_space_token\null}
245     \__algorithme_node:nn {pour}{ALLANT_DE}
246     \__algorithme_node:nn {normal}{\null\c_space_token #2\unskip
247         \c_space_token A~#3\unskip\c_space_token\null}
248     \__algorithme_block:nn {pour} {DEBUT_POUR}
249     #4
250     \__algorithme_block:nn {pour} {FIN_POUR}
251     \__algorithme_unindent:
252 }
253
254 \__algorithme_new_balanced_command:nNNnn {__algorithme_TANTQUE:w} \DEBUTTANTQUE \FINTANTQUE
255     { r() u{\DEBUTTANTQUE} +m }
256 {
257     \__algorithme_indent:nn {tantque} {TANT_QUE}
258     \__algorithme_node:nn {normal}{\null\c_space_token(#1)\c_space_token\null}
259     \__algorithme_node:nn {tantque}{FAIRE}
260     \__algorithme_block:nn {tantque} {DEBUT_TANT_QUE}
261     #3
262     \__algorithme_block:nn {tantque} {FIN_TANT_QUE}

```

```

263     \_algorithme_unindent:
264 }
265
266 \NewDocumentCommand \_algorithme_LINE:w { s O{normal} m } {
267     \IfBooleanTF {#1} {
268         \_algorithme_indent:nn {#2} {#3}
269         \_algorithme_unindent:
270     }{
271         \_algorithme_block:nn {#2} {#3}
272     }
273 }
274
275 \NewDocumentCommand \_algorithme_NODE:w { O{normal} m } {
276     \_algorithme_node:nn {#1} {#2}
277 }
278

```

2.4 User commands and environment

```

279
280
281 \cs_new_protected_nopar:Nn \_algorithme_begincode:n {
282     \int_zero:N \l_algorithme_level_int
283     \begin{tikzpicture}[#1]
284         \coordinate (H) at (0,0);
285         \coordinate (A\int_use:N \l_algorithme_level_int) at (0,0);
286         \cs_set_eq:NN \par \prg_do_nothing:
287         \cs_set_eq:NN \; \prg_do_nothing:
288         \cs_set_eq:NN \LINE \_algorithme_LINE:w
289         \cs_set_eq:NN \NODE \_algorithme_NODE:w
290         \cs_set_eq:NN \VARIABLES \_algorithme_VARIABLES:w
291         \cs_set_eq:NN \DEBUTALGORITHME \_algorithme_DEBUTALGORITHME:w
292         \cs_set_eq:NN \FINALALGORITHME \_algorithme_FINALALGORITHME:w
293         \cs_set_eq:NN \SI \_algorithme_SI:w
294         \cs_set_eq:NN \POUR \_algorithme_POUR:w
295         \cs_set_eq:NN \TANTQUE \_algorithme_TANTQUE:w
296         \cs_set:Npn \AFFICHER {\_algorithme_keyword:nw {AFFICHER}}
297         \cs_set:Npn \AFFICHERCALCUL {\_algorithme_keyword:nw {AFFICHERCALCUL}}
298         \cs_set:Npn \LIRE {\_algorithme_keyword:nw {LIRE}}
299         \cs_set:Npn \PAUSE {\_algorithme_block:nn{motclef}{PAUSE}}
300         \cs_set:Npn \//##1\; {\_algorithme_block:nn{commentaire}{//##1}}
301     }
302 \cs_new_protected_nopar:Nn \_algorithme_closecode: {
303     \end{tikzpicture}
304 }
305
306 \NewDocumentCommand \_algorithme_keyword:nw {m +u{\;}} {
307     \_algorithme_block:nn {motclef} {#1}
308     \_algorithme_node:nn {normal}{\null\c_space_token#2}
309 }
310
311
312 \NewDocumentCommand \smalgorithme { O{ } m } {
313     \group_begin:
314     \_algorithme_begincode:n {every~smalgorithme/.try,#1}

```

```

315     \_algobox_parseall:n {#2}
316     \_algobox_closecode:
317     \group_end:
318 }
319
320 \NewDocumentEnvironment {algobox} {0{}} {
321     \cs_set_protected:Nn \_algobox_algobox:n {
322         \_algobox_begincode:n {every~algobox/.try,#1}
323         \_algobox_parseall:n {##1}
324         \_algobox_closecode:
325     }
326     \Collect@Body\_algobox_algobox:n
327 }{
328 }
329 \end{package}

```