The alg package*

Staffan Ulfberg
staffanu@nada.kth.se

23 September 1997

Abstract

This package defines two environments for typesetting algorithms in LaTeX 2ε. Lines are automatically numbered and can be referenced, the means for easy indentation is provided, and algorithms can be made floating bodies if desired.

1 Introduction

algtab `alg.sty` defines two environments. The `algtab` environment is used to typeset an algorithm with automatically numbered lines; it also takes care of indentation.
algorithm The `algorithm` environment can be used to encapsulate the `algtab` environment in a floating body together with a header, a caption, etc.
The package recognizes a few language options and changes its fixed output strings accordingly. Please tell the author of this package if you need support for another language!

1.1 The `algorithm` environment

The `algorithm` environment starts a floating body; its placement is decided by an optional agrument that can be any combination of `t`, `b`, `p`, or `h`. By default, the placement `H` (Here!) is used, so the float does not float after all. See the description of the float package by Anselm Lingnau for more details about this.
Inside the `algorithm` environment, the text will be indented from the left and right edges. It is possible to put a `\caption` within the float, and also to generate
\listofalgorithms a list of algorithms with the `\listofalgorithms` command.
Notice that it is by no means necessary to use this environment to use the rest of the package.

1.2 Algorithm description

Three macros are defined to describe an algorithm's name, parameters, and function. They may be used in the `algorithm` environment, or just anywhere before the `algtab` environment.

\algname `\algname` takes two arguments of which the first is printed using small caps, followed by the second in parentheses.

*This file was last revised 23 September 1997.

\algdescript `\algdescript` takes one argument which is printed after the string "Description:" in boldface letters.

\alginout `\alginout` takes two arguments which are printed after the strings "Input:" and "Output:" respectively.

1.3 The algtab environment

In the `algtab` environment, an algorithm is typed line by line, separating lines by `\\`. Every time a new line is started, it will be numbered in the left margin starting at line 1. A single "line" of the algorithm can span several lines of the page, but a line number will only be printed after each `\\` command. The current `\ref` value is also set, so that a label can be entered; it's recommended that

\alglabel `\alglabel` is used for this purpose.

algtab* The `algtab` environment has a star-form variant that supresses line number generation; if the `\alglabel` macro is used in this case, the supplied label argument is printed as-is in the left margin of the algorithm (where the line numbers usually

\algref are printed). The macro `\algref` corresponds to `\alglabel` in the way that it prints either the referenced label's value, or the label name literally if no line numbers are used. However, `\algref` can't be used outside the `algtab` environment: use `\ref` intead.

\algnonumber The `\algnonumber` macro can be used at the start of a line to suppress the printing of a line number on that line.

The `algtab` environment indents the algorithm to make room for the line numbers; the `algtab*` does not indent the algorithm by default. However, both environments take an optional argument specifying the amount of indentation that is desired.

\algbegin \algend There are two important macros, `\algbegin` and `\algend` defined in the `algtab` environment. These macros start and stop additional indentation respectively; they should always be used at the beginning of a line. These macros are already included in some of the macros that are special to the `algtab` environment.

2 Example

This section shows how to typeset Algorithm **??** using the alg package in LaTeX 2ε:

```
\begin{algorithm}[h]
\caption{Example of the \texttt{algorithm} environment.\label{alg:ex}}
\alginout{An ordered set $U=\{u_1,u_2,\dots,u_n\}$.}
         {The set's maximum element, and a set $A$, $|A|=\log n$,
          containing the next largest element.  (If $n=1$, then
          $A=\emptyset$).}
\algname{Max2}{$U$}
\begin{algtab}
\algif{$|U|=1$}
        \algreturn $(u_1, \emptyset)$ \\
\algelsif{$|U|=2$}
        \algifthenelse{$u_1>u_2$}{\algreturn $(u_1, \{u_2\})$}
                                 {\algreturn $(u_2, \{u_1\})$}
\algelse
        $(b,B) \leftarrow$
```

Algorithm 1: Example of the `algorithm` environment.
Input: An ordered set $U = \{u_1, u_2, \ldots, u_n\}$.
Output: The set's maximum element, and a set A, $|A| = \log n$, containing the next largest element. (If $n = 1$, then $A = \emptyset$).

Max2(U)
- **if** $|U| = 1$
  - **return** $(u_1, \emptyset)$
- **else if** $|U| = 2$
  - **if** $u_1 > u_2$ **then return** $(u_1, \{u_2\})$
    - **else return** $(u_2, \{u_1\})$
- **else**
  - $(b, B) \leftarrow$ Max2($\{u_i\}_{i=1}^{\lfloor n/2 \rfloor}$)
  - $(c, C) \leftarrow$ Max2($\{u_i\}_{i=\lfloor n/2 \rfloor+1}^{n}$)
  - **if** $b > c$ **then return** $(b, \{c\} \cup B)$
    - **else return** $(c, \{b\} \cup C)$

```
           \algcall{Max2}{$\{u_i\}_{i=1}^{\lfloor n/2\rfloor}$}\\
        $(c,C) \leftarrow$
           \algcall{Max2}{$\{u_i\}_{i=\lfloor n/2\rfloor+1}^{n}$}\\
        \algifthenelse{$b>c$}{\algreturn $(b, \{c\}\cup B)$}
                             {\algreturn $(c, \{b\}\cup C)$}
\algend
\end{algtab}
\end{algorithm}
```

There is a list of all programming constructs available in the `algtab` environment at the end of this document. They work very much like the macros used above. As a general rule, if a macro doesn't take an argument, a space charater is included at the end of the macro definition. For example, type `$a=0$ \algor \algnot $b=0$` to produce "$a = 0$ **or not** $b = 0$."

3 The Implementation

```
⟨*package⟩
```

The `float` package by Anselm Lingnau is used for floating algorithms.

```
\RequirePackage{float, ifthen}
```

Here are the varible declarations. `\algmarginwidth`, `\alglinenowidth`, and `\algtabwidth` define the amount of indentation of the entire algorithm, the space reserved for line numbers, and the indentation made by `\algbegin` respectively.

```
\newlength{\algmarginwidth}\setlength{\algmarginwidth}{.5in}
\newlength{\alglinenowidth}\setlength{\alglinenowidth}{1.2cm}
\newlength{\algtabwidth}\setlength{\algtabwidth}{.5cm}
\newlength{\alg@fromleft}
\newlength{\alg@tmplen}
\newsavebox{\alg@tmpbox}
\newcounter{alg@inmargin}\setcounter{alg@inmargin}{0}
\newcounter{algline}
\newboolean{alg@linenums}
```

```
\newboolean{alg@nonumber}
```

The strings used in captions and in the List of Algorithms differ in some languages.

```
\DeclareOption{english}{\def\alg@floatname{Algorithm}
                        \def\alg@listname{List of Algorithms}
                        \def\alg@descname{Description}
                        \def\alg@inputname{Input}
                        \def\alg@outputname{Output}}
\DeclareOption{american}{\def\alg@floatname{Algorithm}
                        \def\alg@listname{List of Algorithms}
                        \def\alg@descname{Description}
                        \def\alg@inputname{Input}
                        \def\alg@outputname{Output}}
\DeclareOption{swedish}{\def\alg@floatname{Algoritm}
                        \def\alg@listname{Algoritmer}
                        \def\alg@descname{Beskrivning}
                        \def\alg@inputname{Input}
                        \def\alg@outputname{Output}}
\DeclareOption{french}{\def\alg@floatname{Algoritme}
                        \def\alg@listname{Liste des algorithmes}
                        \def\alg@descname{Description}
                        \def\alg@inputname{Entr\'{e}e}
                        \def\alg@outputname{Sortie}}
\ExecuteOptions{english}
\ProcessOptions
```

The following definitions set up the properties of the `algorithmfloat`.

```
\newcommand\floatc@alg[2]{{\bfseries\rmfamily
   \hspace{\algmarginwidth}#1:} #2\par}
\newcommand\fs@alg{
   \let\@fs@capt\floatc@alg
   \def\@fs@pre{}\def\@fs@post{}\def\@fs@mid{\vspace{3pt}}
   \let\@fs@iftopcapt\iftrue}
\floatstyle{alg}
\newfloat{algorithmfloat}{h}{loa}
\floatname{algorithmfloat}{\alg@floatname}
```

`\listofalgorithms` The `\listofalgorithms` macro can be used to generate a list of all algorithms in a document.

```
\newcommand{\listofalgorithms}{\listof{algorithmfloat}{\alg@listname}}
```

`\alg@margin` The `alg@margin` macro makes the text a bit narrower. It is used in the start of both the `algorithm` and `algtab` environments, and also in the `algname`, `algdescript`, and `alginout` macros ; it keeps track of if the text is already narrow, and in this case does nothing.

```
\newcommand{\alg@margin} {
   \ifthenelse{\value{alg@inmargin}=0} {
       \advance\leftskip\algmarginwidth
       \advance\rightskip\algmarginwidth
       \alg@fromleft=\leftskip
   } {}
   \stepcounter{alg@inmargin}
   \parskip=0cm\parindent=0cm
}
```

\alg@unmargin

The alg@unmargin macro resets any indentation made by alg@margin.

```
\newcommand{\alg@unmargin} {
   \setcounter{alg@inmargin}{0}%
   \advance\leftskip-\algmarginwidth%
   \advance\rightskip-\algmarginwidth%
}
```

algorithm

The algorithm environment simply begins a float as defined above. Actually, the default is to use the placement parameter H, so that the algorithm will not really float. Inside the float all text is indented by \algmarginwidth.

```
\newenvironment{algorithm}[1][H] {
   \begin{algorithmfloat}[#1]\alg@margin
} {
   \alg@unmargin\end{algorithmfloat}
}
```

alg@tab

The alg@tab environment is what does most of the formatting job; it's called by algtab and algtab* defined below. The argument defines the amount of initial indentation. If the counter alg@inmargin is zero, alg@tab is not started within a float. In this case some room is made above and below it. Inside this environment, \\ is let to the macro alg@cr that is used to begin new lines. The catcode for ^^M is changed to admit blank input lines within an algorithm.

```
\newenvironment{alg@tab}[1] {
   \setboolean{alg@nonumber}{false}%
   \ifthenelse{\value{alg@inmargin}=0} {\vskip\baselineskip}{}
   \alg@margin
   \let\\=\alg@cr
   \catcode`\^^M=10
   \setcounter{algline}{0}\refstepcounter{algline}
   \advance\leftskip#1
   \alg@putlineno\ignorespaces
} {
   \setbox\alg@tmpbox=\lastbox
   \ifhbox\alg@tmpbox{\vskip-\baselineskip}\else\par\fi
   \alg@unmargin
   \ifthenelse{\value{alg@inmargin}=0}{\vskip\baselineskip}{}
}
```

algtab

This environment sets alg@linenums true, which will make \algputlineno write the current line number in the left margin.

```
\newenvironment{algtab}[1][\alglinenowidth] {
   \setboolean{alg@linenums}{true}\begin{alg@tab}{#1}
} {\end{alg@tab}}
```

algtab*

The algtab* environment works like algtab but sets alg@linenums false and uses zero indentation by default.

```
\newenvironment{algtab*}[1][0cm] {
   \setboolean{alg@linenums}{false}\begin{alg@tab}{#1}
} {\end{alg@tab}}
```

\alg@kill

The \alg@kill macro removes the last box from the current horizontal list. It is used to remove the box containing the line number (label) when indentation is changed: in this case a new box for the label must be created.

```
\newcommand{\alg@kill}{\setbox\alg@tmpbox=\lastbox%
   \ifvoid\alg@tmpbox\PackageError{alg}{Attempt to remove label
      in middle of line}\fi}
```

`\algbegin` \algbegin adds to \leftskip and replaces the line number. It must only be used at the beginning of a line.

```
\newcommand{\algbegin}[1][\algtabwidth]{\advance\leftskip#1%
   \alg@kill\alg@putlineno\ignorespaces}
```

`\algend` This macro reverses the effect of a previous \algbegin.

```
\newcommand{\algend}[1][\algtabwidth]{\advance\leftskip-1#1%
   \alg@kill\alg@putlineno\ignorespaces}
```

`\algnonumber`

```
\newcommand{\algnonumber}{\alg@kill\alg@putlabel{}%
   \setboolean{alg@nonumber}{true}\ignorespaces}
```

`\alg@cr` New lines are started using this macro. The line number is incremented and printed.

```
\newcommand{\alg@cr}{\par\refstepcounter{algline}%
   \setboolean{alg@nonumber}{false}\alg@putlineno\ignorespaces}
```

`\alg@putlineno` Line numbers are typeset as labels.

```
\newcommand{\alg@putlineno} {%
   \ifthenelse{\boolean{alg@linenums}} {%
      \ifthenelse{\boolean{alg@nonumber}}{\alg@putlabel{}} {%
         \alg@putlabel{(\arabic{algline})}}}%
      {\alg@putlabel{}}}
```

`\alg@putlabel` This is the macro that puts a label on the horizontal list. The label extends to the left of a zero-width box.

```
\newcommand{\alg@putlabel}[1]{{%
      \alg@tmplen=\leftskip \advance\alg@tmplen-\alg@fromleft%
      \makebox[0cm][r]{\makebox[\alg@tmplen][l]{#1}}}}
```

3.1 Macros for algorithm descriptions

```
\newcommand{\algdescript}[1]{\alg@margin\textbf{\alg@descname: }#1\par}
\newcommand{\alginout}[2]{\alg@margin\textbf{\alg@inputname: }#1\par
   \textbf{\alg@outputname: }#2\par}
\newcommand{\algname}[2]{\alg@margin\textsc{#1}(#2)\par}
```

3.2 Macros for referencing

```
\newcommand{\alglabel}[1]{%
   \ifthenelse{\boolean{alg@linenums}}{%
      \label{#1}}{\alg@kill\alg@putlabel{#1}}\ignorespaces}
\newcommand{\algref}[1]{\ifthenelse{\boolean{alg@linenums}}%
   {\ref{#1}}{#1}}
```

3.3 Macros for the algtab environment

```
\newcommand{\algand}{\mbox{\textbf{and }}}
\newcommand{\algbreak}{\textbf{break}}
```

```
\newcommand{\algcall}[2]{\textsc{#1}(#2)}
\newcommand{\algcase}[1]{\algend\textbf{case} #1\\\algbegin}
\newcommand{\algcontinue}{\textbf{continue}}
\newcommand{\algdefault}{\algend\textbf{default}\\\algbegin}
\newcommand{\algelse}{\algend\textbf{else}\\\algbegin}
\newcommand{\algelseif}[1]{\algelsif{#1}}
\newcommand{\algelsif}[1]{\algend\textbf{else if} #1\\\algbegin}
\newcommand{\algerror}{\textbf{error }}
\newcommand{\algfalse}{\mbox{\textbf{false }}}
\newcommand{\algforto}[2]{\textbf{for} #1 \textbf{to} #2\\\algbegin}
\newcommand{\algforeach}[1]{\textbf{foreach} #1\\\algbegin}
\newcommand{\alggoto}{\textbf{goto~}}
\newcommand{\algif}[1]{\textbf{if} #1\\\algbegin}
\newcommand{\algifthen}[2]{\textbf{if} #1 \textbf{then} #2\\}
\newcommand{\algifthenelse}[3]{\setbox\alg@tmpbox=
  \hbox{\textbf{if} #1 }\copy\alg@tmpbox\textbf{then} #2\\
  \settowidth{\alg@tmplen}{\box\alg@tmpbox}%
  \algbegin[\alg@tmplen]\textbf{else} #3\\ \algend[\alg@tmplen]}
\newcommand{\algnot}{\mbox{\textbf{not }}}
\newcommand{\algor}{\mbox{\textbf{or }}}
\newcommand{\algpardo}{\mbox{\textbf{pardo}}}
\newcommand{\algprint}{\textbf{print }}
\newcommand{\algrepeat}{\textbf{repeat}\\\algbegin}
\newcommand{\algreturn}{\textbf{return~}}
\newcommand{\algswitch}[1]{\textbf{switch} #1\\\algbegin}
\newcommand{\algtrue}{\mbox{\textbf{true }}}
\newcommand{\alguntil}[1]{\algend\textbf{until} #1\ \\}
\newcommand{\algwhile}[1]{\textbf{while} #1\\\algbegin}
```

⟨/package⟩