**ifallfalse** – Compare string against set of strings

Dennis Chen  
proofprogram@gmail.com  
v2.0.0, 2021/07/22

Abstract  
The **ifallfalse** package is a package that allows you to check whether a string is contained within another set of strings, and perform an action if it is not.

1 Usage  
The package provides an **allfalse** environment and a macro \texttt{orcheck} to be used inside the **allfalse** environment.

To set up an **allfalse** environment, simply write

\begin{allfalse}{string}{true branch}{false branch}
\end{allfalse}

\texttt{string} will be compared to the set of strings (which we will declare via \texttt{orcheck}), and if \texttt{string} does not match the set of strings, \texttt{false branch} will be executed. Otherwise, \texttt{true branch} will be executed.

To add strings to the set that \texttt{string} will be compared to, we must write \texttt{orcheck(setstring)} inside the corresponding **allfalse** environment. Then, \texttt{action} will not execute if \texttt{string} matches \texttt{setstring} or any arguments of previous \texttt{orcheck} declarations.

If no \texttt{orcheck} declarations exist, then \texttt{action} will always be executed.

1.1 Error Checking  
The package checks whether the macro \texttt{orcheck} is used inside an **allfalse** environment. If it is not, the package throws an error.

\*https://github.com/chennisden/ifallfalse
2 Example

Here is a simple example to demonstrate how allfalse is used.

\documentclass{minimal}
\usepackage{ifallfalse}
\begin{document}
\begin{allfalse}{purple}{}{This color is not red, blue, or green!}
  \orcheck{red}
  \orcheck{blue}
  \orcheck{green}
\end{allfalse}
\end{document}

In this case, because purple does not match red, blue, or green, the false branch — which is This color is not red, blue, or green! — will execute at that location inside the document.

3 Implementation

These are the implementation details of package allfalse. Because the package is so short, we can explain everything.

allfalse When setting up allfalse, we locally define the \comparedstring macro with the first argument that the environment takes in. This is what will be compared against all the strings passed in through the \orcheck declarations inside the environment.

Then, we define our body of logic (which we will be adding onto through \orcheck) to just initially consist of the action we would like to perform if \comparedstring matches none of the strings passed in through \orcheck.

Finally, we execute our logicbody, which will change \ifallfalse@branch to be false if appropriate. Then, the appropriate action will be executed.

\begin{verbatim}
1 \newenvironment{allfalse}[3]
  2 {
  3 \newif\ifallfalse@branch\allfalse@branchtrue%
  4 \def\comparedstring{#1}%
  5 \def\trueaction{#2}%
  6 \def\falseaction{#3}%
  7 \def\logicbody{\allfalse@branchfalse}%
  8 }
  9 {
 10 \logicbody%
 11 \ifallfalse@branch
 12 \trueaction%
\end{verbatim}
\texttt{\textbackslash orcheck} We first save \texttt{allfalse} to a macro so we can use \texttt{\ifx} to compare the current environment name against it. If we can, then we add some following (somewhat convoluted) code to \texttt{\logicbody}. I will explain what each piece of it does, though not in the order the pieces of code appear.

- \texttt{\ifx\@currenvir\@allfalsename} evaluates to true if the current environment (whose name is saved to the macro \texttt{\@currenvir}) matches the name of \texttt{\@allfalsename}, or \texttt{allfalse}.
- If it evaluates to \texttt{false}, the package throws an error.
- The line \texttt{\pdfstrcmp{\comparedstring}{#1}=0} evaluates to true when put with \texttt{\ifnum} if the two arguments passed into \texttt{\pdfstrcmp} are equal, because \texttt{pdfstrcmp} compares their lexicographical order and returns \texttt{0} if the two strings are lexicographically equivalent.
- Thus, we can treat \texttt{\ifnum\pdfstrcmp{\comparedstring}{#1}=0} as an expression that evaluates to true if \texttt{\comparedstring} and \texttt{#1} match, and \texttt{false} otherwise.
- When all is said and done, the logic reduces to something of the form

\begin{verbatim}
\if\else
\if\else
\ldots \allfalse@branchfalse
\fi\ldots \fi
\end{verbatim}

Logically, \texttt{\allfalse@branchfalse} will only execute if all the conditions are \texttt{false}; in other words, it will only execute if \texttt{\comparedstring} does not match any of the strings passed in via \texttt{\orcheck}. This is because each \texttt{\else} branch must execute.

\begin{verbatim}
17 \newcommand*{\@allfalsename}{allfalse}
18 \newcommand{\orcheck}[1]{
19 \ifx\@currenvir\@allfalsename
20 \protected@edef\logicbody{
21 \ifnum\pdfstrcmp{\comparedstring}{#1}=0\else\logicbody\fi
22 }
23 \else
24 \PackageError{ifallfalse}{
25 \protect\orcheck\space should be nested within the allfalse environment
26 }{}
27 \fi
28 }
29 }
\end{verbatim}
4 Limitations

This package cannot be used in the lualatex engine.