Axessibility: Creating PDF documents with accessible formulae

Axessibility: creating PDF documents with accessible formulae

D. Ahmetovic, T. Armano, M. Berra, C. Bernareggi, A. Capietto, S. Coriasco, N. Murru and A. Ruighi

Abstract

PDF documents containing formulae generated by LaTeX are usually not accessible by assistive technologies for visually impaired people (i.e., by screen readers and braille displays). The LaTeX package `axessibility.sty` that we developed manages this issue, allowing to create PDF documents where the formulae are read by these assistive technologies, since it automa tically generates hidden comments in the PDF document (using the `/ActualText` attribute) in correspondence to each formula. This actual text is hidden in the PDF document, but the screen readers JAWS, NVDA and VoiceOver read it correctly. Moreover, we have created NVDA and Jaws dictionaries (in English and in Italian) that provide the reading in the natural language in the case that the user does not know the LaTeX commands. The package does not generate PDF/UA.

1 Introduction

In this paper, we describe `axessibility.sty`, a LaTeX package which allows to automatically generate a PDF document with formulae accessible by assistive technologies for visually impaired people. Assistive technologies (screen readers and braille displays) perform satisfactorily with regard to di gital documents containing text, but they still have a long way to go as far as formulae and graphs are concerned. A comprehensive overview about this problem can be found in [1] and [3].

Many studies have been conducted in order to improve the accessibility of digital documents with mathematical contents. For instance, MathPlayer ensures accessibility of formulae inserted by using MathType in Word documents [12]. Another way for creating accessible mathematical documents is given by the MathML language (see [6] for further information). However, accessibility of such documents is heavily affected by the versions of browsers, operating syste ms and screen readers, making this solution very unstable. A system used by blind people for reading and writing mathematics is the LAMBDA system (Linear Access to Mathematics for braille Device and Audio-synthesis). Mathematical language in LAMBDA is designed so that every symbol can be directly translated into words. For further details on LAMBDA we refer to [5]. Unfortunately, this system does not help to spread accessible digital documents, since it is only used by visually impaired people and it is not a standard for the realization of documents by sighted people. Regarding LaTeX, assistive technologies can directly manage LaTeX documents. In this case, visually impaired people need to learn LaTeX in order to understand the commands. However, there are software which facilitate LaTeX comprehension and usability; one of them is BlindMath [11]. Moreover, some converters from LaTeX to braille exist, see, e.g., [10] and [4].

In general, the most widespread digital documents are in PDF format. How ever, in the case of mathematical contents, they are not accessible at all, since formuale are usually unreadable by screen readers because they are bidimensional as images. None of the above systems allows to directly produce accessible formulae in PDF documents. This could be possible only performing specific tasks. For instance, using the Word editor, if each formula is manually tagged by the author (by using the alternative text), such a comment will be kept when the corresponding PDF file will be generated and it will be read by the screen reader. However, this procedure does not help to improve the presence of accessible PDF documents, since it is a very boring and time consuming method. It is very hard to think that an author performs these actions for the realization, e.g., of a book. Currently, a standard and fast method for inserting accessible formulae into a PDF documents is still lacking despite it is a very important issue for spreading accessible digital scientific documents. In [13] standard guidelines for accessibility of PDF documents are presented. Moreover, in [8], [9] and [7], an overview about accessibility of PDF documents is provided with a focus on mathematical contents.

In this paper, we show the features of the package `axessibility.sty` (whose a first version is also described in [2]) that provides the first method for an automatized production of accessible PDF documents with mathematical contents. We would like to highlight that this package does not produces fully tagged PDF, such as the standard PDF/UA , but it allows to obtain a PDF where formulae are described using the `/ActualText` attribute.

2 Problem Statement

When a PDF document is generated starting from LaTeX, formulae are no t accessible by screen readers and braille displays. They can be made accessible by inserting a hidden comment, i.e., an actual text, similarly to the case of web pages or Word documents. This can be made, e.g., by using the LaTeX package pdfcomment.sty or using an editor for PDF files like

Adobe Acrobat Pro. In any case, this task must be manually performed by the author and it is surely inefficient, since the author should write the formulae and, in addition , insert a description for each formula. Note also that the package `pdfcomment.sty` does not allow to insert special characters like *backslash*, *brace*, etc, in the comment. Moreover, with these solutions, the reading is bothered, since the screen reader reads incorrectly the formula and then the correct comment of the formula.

In Figure 1, we show the PDF document generated from the following LaTeX code containing a simple formula with a comment manually inserted.

```
\documentclass{article}
\documentclass[a4paper]{article}
\usepackage{pdfcomment}
\begin{document}
A simple formula:
\begin{equation}
\pdftooltip{
        \frac{1 + \sqrt{5}}{2}
        }{
        begin fraction numerator 1 +
    square root of 5 over 2 end fraction
        }
\end{equation}
\end{document}
```

When the screen reader accesses the PDF document, the formula will be read

> square root 1 plus 5 2 begin fraction numerator 1 plus square root of 5 over 2 end fraction

i.e., before reading the correct comment

> begin fraction numerator 1 plus square root of 5 over 2 end fraction

the screen reader reads incorrectly the formula

> square root 1 plus 5 2.

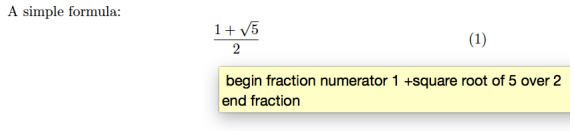

Figure 1: PDF document generated using the package pdfcomm ent.sty

There are also some LaTeX packages that try to improve the accessibility of PDF documents produced by LaTeX. In particular the packages `accsupp.sty` and `accessibility.sty` — available at `https://ctan.org/pkg/accsupp` and `https://github.com/AndyClifton/AccessibleMetaClass` respectively — has been developed in order to obtain tagged PDF documents. However, both packages do not solve the problem of the accessibility of formulae.

3 `axessibility.sty` LaTeX package

Our package, named `axessibility.sty`, solves the problem described in the previous section. It achieves that by inserting a hidden comment in the PDF file corresponding to any given formula. This comment, named `/ActualText`, contains the original LaTeX commands used to generate the formula. The hidden comment is read by screen readers and braille displays instead of the `ASCII` representation of the formula, which is often incorrect.

3.1 Usage

Authors that would like to create an accessible PDF document for visually impaired people only need to include the `axessibility.sty` package into the preamble of their LaTeX project. Mathematical environments automatically produce the `/ActualText` content and include it in the produced PDF file.

We have treated the most used environments for inserting formulae, i.e., `equation`, `equation*`, `\[`, `\(`. Hence, any formula inserted using one of these environments is accessible in the corresponding PDF document. Additionally, the package enables to copy the formula LaTeX code from the PDF reader and paste it elsewhere.

Note that, to preserve the compatibility with Acrobat Reader, our package disco urages the use of the underscore character _ which is not correctly r ead using screen readers in combination with this PDF reader. Alternativ ely, we suggest to use the equivalent command `\sb`.

In-lined and display mathematical modes (`$`, `$$`) are not supported in this version of the package. However external scripts provided as companion software can also address these use cases.

If we use the package `axessibility.sty` applied to the previous example , we obtain the following LaTeX code.

```
\documentclass[a4paper,11pt]{article}
\usepackage{axessibility}
\begin{document}
    A simple formula:
    \begin{equation}
        \frac{1 + \sqrt{5}}{2}
    \end{equation}
\end{document}
```

We observe that, in this case, the author has to write the formula without adding anything else. Morevoer, inside the source code of the PDF file, we find an `/ActualText` tag with the LaTeXcode inside, automatically generated by the `axessibility.sty` package.

```
/S/Span<</ActualText(\040\040\\frac
\040{1\040+\040\\sqrt
\040{5}}{2}\040)
>>
BDC
```

The screen reader will read correctly the LaTeX command `\frac {1+ \sqrt {5}}{2}`. Moreover, we have created JAWS and NVDA dictionaries that provide the reading in the natural language in the case that the user does not know the LaTeX commands.

3.2 Technical Overview

`axessibility.sty` first defines a pair of internal commands (`\BeginAxessible` and `\EndAxessible`) which redefine `\BeginAccSupp` and `\EndAccSupp` as follows:

```
\newcommand*{\BeginAxessible}[1]{%
  \begingroup
    \setkeys{ACCSUPP}{#1}%
    \edef\ACCSUPP@span{%
     /S/Formula<<%
        \ifx\ACCSUPP@Alt\relax
        \else
          /Alt\ACCSUPP@Alt
        \fi
        \ifx\ACCSUPP@ActualText\relax
        \else
          /ActualText\ACCSUPP@ActualText
        \fi
      >>%
    }%
    \ACCSUPP@bdc
    \ACCSUPP@space
  \endgroup
}
```

Precisely, `\BeginAxessible` adds a hidden comment that starts with `/S/Formula` instead of `/Span`.

```
\newcommand*{\EndAxessible}{%
  \begingroup
    \ACCSUPP@emc
  \endgroup
}
```

The second building block of this package is the wrapper. This routine takes the LaTeX code inside the formula, removes the tokens and passes it to `\BeginAxessible`.

```
\long\def\wrap#1{
\BeginAxessible{method=escape,
ActualText=\detokenize\expandafter{#1},
Alt=\detokenize\expandafter{#1} }
#1
\EndAxessible%
}
```

Finally, using the wrapper, we can re-define the mathematical environments using the command above. Here is an example using `equation`.

```
\renewenvironment{equation}{%
 \incr@eqnum
  \mathdisplay@push
  \st@rredfalse \global\@eqnswtrue
 \mathdisplay{equation}%
  \collect@body\wrap\auxiliaryspace}{%
  \endmathdisplay{equation}%
  \mathdisplay@pop
  \ignorespacesafterend
}
```

4 Conclusions and Future Work

We have developed a LaTeX package that automatically generates comments to formulae when the PDF document is produced by LaTeX. The comments are hidden in the PDF document and they contain the LaTeX commands that generate the formulae. In this way, an accessible PDF document containing formulae is generated. Indeed, screen readers are able to access the comment when processing a formula and reading it. Moreover, we have created JAWS and NVDA dictionaries that provides the reading in the natural language in the case that the user does not know the LaTeX commands.

There are a few issues that are yet to be solved with a pure LaTeX solution. Namely,

- Mathematical environments delimited with `$`, `$$`,
- User-defined Macros,
- Multi-line environments such as `\align` and `\eqnarray`.
- Semantic description of formulae.
- PDF/UA.

We have resolved the first two problems using an external script – `axesscleaner.py` – coded in `Perl` and `Python`. The script also beautifies the LaTeX file and removes all the *underscore* characters _, replacing those with `\sb`. Using this solution we are now able

to apply `axessibility.sty` to entire textbooks that were written without using the package in the first place.

Multi-line environments are going to be treated using a LaTeX solution that is currently in the test phase. Concerning the last two problems a more in-depth research is in order. The authors are currently initiating the investigation to address these issues as a future work.

5 Acknowledgements

The authors wish to thank the bank foundation 'Fondazione Cassa di Risparmio di Torino', LeoClub (Biella, Italy) and the several volunteers with visual impairment who provided their fundamental contribution.

References

[1] D. Archambault, B. Stoger, et al. Access to scientific content by visually impaired people. *Upgrade* 2:14, 2007.

[2] T. Armano, A. Capietto, et al. An automatized method based on latex for the realization of accessible pdf documents containing formulae. *Computers Helping People with Special Needs, Lecture Notes in Computer Science* 10896:583–589, 2018.

[3] T. Armano, A. Capietto, et al. An overview on ict for the accessibility of scientific texts by visually impaired students. *Proceedings Conference SIREM-SIE-L* 2014:119–122, 2014.

[4] M. Batusic, S. Miesenberger, K., and B. LaBraDoor. a contribution to making mathematics accessible for the blind. In *Proceedings of 6th International Conference on Computers Helping People with Special Needs*, Oldenbourg, Wien, Munchen, 1998. ICCHP 1998.

[5] C. Bernareggi. Non-sequential mathematical notations in the lambda system. *Computers Helping People with Special Needs, Lecture Notes in Computer Science* 6180:389–395, 2010.

[6] C. Bernareggi and D. Archambault. Mathematics on the web: emerging opportunities for visually impaired people. In P. W. 4A., ed., *07 Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A), Banff (Canada),*, pp. 108–111, 2007.

[7] M. Borsero, N. Murru, and A. Ruighi. Il latex come soluzione al problema dell'accesso a testi con formule da parte di disabili visivi. *ArsTeXnica* 22:12–18, 2016.

[8] R. Moore. Ongoing efforts to generate tagged pdf using pdftex. *TUGboat* 30:170–175, 2009.

[9] R. Moore. Pdf/a-3u as an archival format for accessible mathematics. In M. Stephen, J. H. Davenport, et al., eds., *Watt*, pp. 184–199. CICM, Lecture Notes in Computer Science 8543, 2014.

[10] A. Papasalouros and A. A. Tsolomitis. direct tex-to-braille transcribing method. *Journal of Science Education for Students with Disabilities* 20(5), 2017.

[11] A. Pepino, C. Freda, et al. Blindmath. In C. Helping, ed., *a New Scientific Editor for Blind Students*, pp. 1171–1174. People with Special Needs, Lecture Notes in Computer Science 4061, 2006.

[12] N. M. w.-b. m. a. Soiffer. Assets '05: Proceedings of the 7th international acm sigaccess conference on computers and accessibility. *New York (USA)* pp. 204–205, 2005.

[13] A. Uebelbacher, R. Bianchetti, and M. P. Riesch. Accessibility checker (pac 2): The first tool to test pdf documents for pdf/ua compliance. *Computers Helping People with Special Needs, Lecture Notes in Computer Science* 8547:197–201, 2014.

◇ D. Ahmetovic
Dipartimento di Matematica "G. Peano", Universit degli Studi di Torino
dragan.ahmetovic@unito.it

◇ T. Armano
Dipartimento di Matematica "G. Peano", Universit degli Studi di Torino
tiziana.armano@unito.it

◇ M. Berra
Dipartimento di Matematica "G. Peano", Universit degli Studi di Torino
michele.berra@unito.it

◇ C. Bernareggi
Dipartimento di Informatica, Universit di Milano
cristian.bernareggi@unimi.it

◇ A. Capietto
Dipartimento di Matematica "G. Peano", Universit degli Studi di Torino
anna.capietto@unito.it

◇ S. Coriasco
Dipartimento di Matematica "G. Peano", Universit degli Studi di Torino
sandro.coriasco@unito.it

◇ N. Murru
Dipartimento di Matematica "G. Peano", Universit degli Studi di Torino
nadir.murru@unito.it

◇ A. Ruighi
Dipartimento di Matematica "G. Peano", Universit degli Studi di Torino
alice.ruighi@unito.it