Aleksandar Pejović; Žarko Mijajlović
Conversion of TeX Documents to PDF

In: Petr Sojka (ed.): Towards a Digital Mathematics Library. Grand Bend, Ontario, Canada, July 8-9th, 2009. Masaryk University Press, Brno, 2009. pp. 121--124.

Persistent URL: http://dml.cz/dmlcz/702550

Conversion of TEX Documents to PDF

Aleksandar Pejović[1] and Žarko Mijajlović[2]

[1] Mathematical Institute SANU, Belgarde, Serbia
pejovica@mi.sanu.ac.rs
[2] Univ. of Belgrade, Faculty of Mathematics, Belgrade, Serbia
zarkom@matf.bg.ac.rs

Abstract. We discuss in some detail some of the drawbacks of PDF files obtained from mathematical papers prepared in TEX, particularly concerning indexing, copy/paste and OCR capabilities.

1 Introduction

We utilized TEX in the nineties of the past century for the digitization and archiving of the journal *Publications de l'Institut Mathématique*, published by the Mathematical Institute in Belgrade, http://elib.mi.sanu.ac.rs. The advantage was the compactness of the archive and the perfect quality of digitized copies. Since then the archive has been constantly enlarged by adding new volumes. The retro-digitization of the journal was completed in 2006 by scanning old volumes published since 1932, when the journal was founded. Technical aspects of the digitization project, can be found in [1] and [2]. In short we found that PDF is a good solution for creating archives of electronic documents, but that some data standards must be obeyed and metadata included into the documents for this purpose [10], because simple conversion to PDF may degrade the use of so obtained PDF documents. The aim of this paper is to discuss several drawbacks in using TEX typesetting system for the retro-digitization that we have encountered recently.

2 Analysis and Solution

After completing retro-digitization, the Institute decided to make the journal repository freely available via Internet. We prepared Internet presentation of our repository (http://publications.mi.sanu.ac.rs) which allowed online browsing, searching and downloading of full texts in PDF. After a while we added it to Google's index with the Google Webmaster Tools [3]. Our aspiration was to take the full advantage of Google PDF files indexing in order to increase the visibility of the content of the journal at the Internet. We accomplished that by assigning unique URL address to every paper. In this way we achieved the following. First, every paper in the repository can be fully indexed and searchable via Google, opposed to our local search. Secondly, hyperlinks to journal papers from our repository should appear in the results of search queries performed by Google. But our surprise was great when we found,

Petr Sojka (editor): DML 2009, Towards a Digital Mathematics Library, pp. 121–124.
 ISBN 978-80-210-4781-5

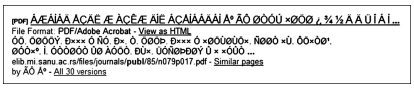
[PDF] ÃÆÁÍÀÃ ÃÇÃÉ Æ ÀÇÊÆ ÃÍË ÁÇÃÍÃÃÃÍ Ã° ÃÔ ØÖÖÚ ×ØÖØ ¿ ¾ ½ Ã Ã Ü Í À Í ...
File Format: PDF/Adobe Acrobat - View as HTML
ÖÖ. ÖØÖÖÝ. Ð»×× Ö ÑÖ. Ð». Ö. ÖØÖÞ. Ð»×× Ö ×ØÚüèÚÖ×. ÑØØÐ ×Ú. ÖÖ×ÖØ¹.
ØÖÖ×* Í ÖÖÖØÖÖ ÙØ ÃÖÖÖ. ÐÙ». ÙÖÑØ»ÞØÝ Ú » ×ÖÚÖ ...
elib.mi.sanu.ac.rs/files/journals/**publ**/85/n079p017.pdf - Similar pages
by ÃÖ Ã° - All 30 versions

Fig. 1: Unrecognizable text

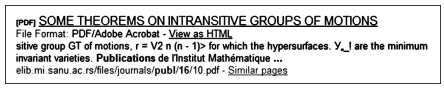
[PDF] SOME THEOREMS ON INTRANSITIVE GROUPS OF MOTIONS
File Format: PDF/Adobe Acrobat - View as HTML
sitive group GT of motions, r = V2 n (n - 1)> for which the hypersurfaces. Y_| are the minimum invariant varieties. Publications de l'Institut Mathématique ...
elib.mi.sanu.ac.rs/files/journals/**publ**/16/10.pdf - Similar pages

Fig. 2: Example of scanned paper

almost accidentally, that the result of indexing was not as we have expected it. Namely, for certain articles Google returns unrecognizable sequences of characters as in the Figure 1. On the other side, Google have worked well for scanned papers, due to built in OCR in Google's indexing engine, Figure 2. We decided to analyze in detail the existing PDF files generated from TEX sources and to find out what went wrong. Many authors practiced similar research for other purposes, for example for extracting mathematical content of PDF documents [4]. Due to the large number of files (1,000), we performed our analysis programmatically. As a result we divided the problematic files into two groups.

Group 1. PDF files with embedded Type 3 fonts, but without ToUnicode CMaps [5]. These files were completely unsuccessfully indexed by Google. The results were similar to those presented in Figure 1.

Group 2. PDF files without textual information. These files were consisted of separated pictures of the characters in CCITT[3] facsimile encoding, not of the codes of characters themselves. Nevertheless, files of this type were indexed by Google, due to embedded OCR technology in the Google indexing engine. Independently of indexing, we consider that PDF files obtained in this way from TEX sources are unacceptable, since they suffer from poor display quality.

Our analysis also shows that PDF copies were obtained from TEX sources using several different methods and software, including: PDF versions 1.2, 1.3, 1.4 and 1.6, dvipdfm ver. 0.13.2, dvips ver. 5.83, 5.90a and 5.95b, Acrobat Distiller 4.05, 5.0 and 6.0, and Adobe Acrobat ver. 7.0, PostScript, Metafont, etc. TEX files have been transformed into PDF format for about fifteen years. This explains why so many different kinds of software have been used; they were changing fast in this period. However, the usage of so many different methods is not acceptable. People who were doing conversion did not care much about configuring the software components in the right way, so the chances for errors were great. One explanation why all these issues remained unnoticed until now would be that PDF files were used then mainly for printing purpose, not for archiving. According to our remarks, we consider that it is important to define a method for converting TEX files to PDF format with advices how to configure accompanying software components. For this purpose, if PDF files are to be used online, we must keep several additional factors in mind: **a.** Search and find capabilities, **b.** Copy and paste results, **c.** Display quality.

[3] The CCITT encoding standard is defined by the International Telecommunications Union (ITU), recommendations T.4 and T.6.

Now we will consider those issues of TEX processing and conversion to PDF that would lead to the output good for distribution and indexing. Even if we have chosen the right fonts, we can still have problems with Internet search and copy/paste capability. The reason lies in PDF which could be considered as the final format for printing. There are no paragraphs and similar constructs, but only the isolated text lines. Hence the search for phrases and hyphenated words might fail. Additional errors may arise from the way how the bounds between words are determined since they are affected by separator characters and by the spacing between them. In TEX and PDF, the position in the plane of every glyph can be controlled. This is one of the most important capabilities of TEX for processing mathematical formulas but in PDF files this capability makes the text searching nontrivial task. But we will not further discuss this matter here.

The problem of text extracting from PDF files is mostly reduced to the font usage and encodings. If PDF files are intended for usage by only one application problems should not be expected, as the encodings are usually embedded in the application itself. But if PDF files are produced for several applications, e.g. Internet searching, the Unicode should be used. There are two aspects of the language problem in PDF documents: showing a text and determination of text content. The former is successfully solved by TEX tools, pdfTeX, dvips and dvipdfm by embedding of needed fonts. The latter is still open for Cyrillic and many other languages. To solve the latter problem one should embed ToUnicode CMap along with the font. This CMap relates codes of font's glyphs to Unicode codes; it can be easily made when such relations are known [6]. According to PDF Reference [7] one can use the following methods to map a character code to a Unicode value:

- If the font dictionary contains a ToUnicode CMap use that CMap to convert the character code to Unicode,
- If the font is a simple/composite font that uses one of the predefined encodings/CMaps use this information to retrieve the corresponding Unicode value (this was very simplified).

If these methods fail to produce a Unicode value, there is no way to determine what the character code represents. So, it is necessary to embed ToUnicode CMap along with embedded font programme[4]. The last statement is confirmed in the example presented in Figure 1. The papers which Google indexed erroneously were converted from TEX to PDF format using Type 3 fonts with non-standard encodings. Without ToUnicode CMaps, Google could not extract texts from these papers. Therefore the use of ToUnicode CMaps is the safest way for full text indexing. It also solves the ligatures problem, i.e. it automatically transforms the selected ligature into separated characters, for example ffi into f f i.

[4] Data for ToUnicode CMap generation can be obtained from a separate file containing code-to-code relations. This file can be connected with corresponding font like encoding file in font dictionary inside a PDF.

The display quality is simply solved by the usage of Type 1 fonts. They are scalable in contrast to Type 3 fonts which are bitmapped. PDF files intended for multiple usages should have embedded Type 1 fonts with appropriate ToUnicode CMaps. Our choice for TEX processing and conversion to PDF is pdfLATEX which is included into the most of new LATEX distributions. We decided to use it in the future and to re-TEX old TEX papers of our journal that were wrongly indexed by Google. This tool is a fairly complete system and it embeds Type 1 fonts in documents by default. The choice of proper fonts is also very important. We use Latin Modern fonts which also come together with newly LATEX distributions. At the Internet site CTAN [8] there is a package **cmap** [9], developed by V. Volovich, which for fonts used in PDF files embeds appropriate ToUnicode CMaps. The package **cmap**, together with Latin Modern fonts is probably the most important part for TEX processing with intention to convert the files to PDF. Additional reason is that they support most of the font encodings (OT1, T1, T2A, T2B, T2C ...). This support is important if one processes multilingual and multi-alphabetic texts. The following preamble should be included in the TEX source in order to use the above mentioned technologies (e.g. this document).

```
\documentclass{llncs}
\usepackage[resetfonts]{cmap}
\usepackage{lmodern}  \usepackage[T1]{fontenc}
```

This is the basic form of preamble that we are using for generating PDF documents, since it assures search and find capabilities, copy/paste results and display quality. The suggested technologies are quite appropriate for the future use, but we expect some problems in automating the process of rebuilding PDF files from old TEX sources. This is due to appearances of various styles, macros and possible several languages and alphabets in a single document.

References

1. Ž. Mijajlović, Z. Ognjanović, *Digitization of Mathematical Editions in Serbia*, In: P. Sojka (ed.) Proceedings of DML 2008, Birmingham, UK, July 27th.
2. Ž. Mijajlović, Z. Ognjanović, A. Pejović, *Internet presentations of mathematical works in Serbia*, NCD Review vol. 12, 2008, 43–48, `http://elib.mi.sanu.ac.rs/`.
3. Google Webmaster Tools, `http://www.google.com/webmasters/tools/`.
4. J.B. Baker, A.P. Sexton, V. Sorge, *Extracting precise data on the mathematical content of PDF documents*, ibid, 75–79.
5. Adobe CMap and CIDFont Files Specification, `http://www.adobe.com/devnet/font/pdfs/5014.CIDFont_Spec.pdf`.
6. ToUnicode Mapping File Tutorial, `http://www.adobe.com/devnet/acrobat/pdfs/5411.ToUnicode.pdf`.
7. PDF Reference 1.7, 6th Ed., `http://www.adobe.com/devnet/pdf/`.
8. The Comprehensive TEX Archive Network, `http://www.ctan.org/`.
9. The cmap package developed by V. Volovich, `http://tug.ctan.org/tex-archive/macros/latex/contrib/cmap/`.
10. PDF as a standard for archiving, white paper, `http://www.adobe.com/enterprise/pdfs/pdfarchiving.pdf`