Преглед НЦД 15 (2009), 19–26

Aleksandar Pejović
Matematički institut SANU
Beograd
Žarko Mijajlović
Matematički fakultet
Beograd

CONVERSION OF TEX GENERATED DOCUMENTS TO PDF FILE FORMAT

Abstract. Mathematically oriented texts are produced mainly by using TEX processing system. In the recent years, PDF has become de facto the standard for archiving, document presentation and exchange on Internet. The conversion from TEX to PDF format is the critical and most important point in obtaining scientific papers in PDF formats. We discuss in this paper some drawbacks of PDF files obtained in this way, particularly concerning indexing, copy/paste and OCR capabilities. We also discuss some shortcomings in uses of PDF files with e-book readers.

1 Introduction

TEX is a powerful text processing language and is the required format for many periodicals now, particularly in mathematical sciences. However, there are other uses of TEX. For example, we utilized TEX in the nineties of the past century for the retro-digitization and archiving of the journal *Publications de l'Institut Mathématique*, published by the Mathematical Institute of the Serbian Academy of Sciences and Arts. Our first archiving technique consisted of the retyping articles using TEX, starting with the most recent issues. About 25 volumes, published between 1980 and 1995, were electronically archived in this way. The main advantage was that the archive was very compact, having less than 100 MB and, of course, the perfect quality of digitized copies. The archive includes TEX-source and output (device independent - DVI, and PDF) files. Since then the archive has been constantly enlarged by adding new volumes and now it contains nearly 60 volumes in TEX format, published between 1980 and 2009. The retro-digitization of the journal was completed in 2006 by scanning old volumes published since 1932, when the *Publications* were founded. The complete digital archive of the journal can be found in the *eLibrary of the Mathematical Institute*, http://elib.mi.sanu.ac.rs/. The technical aspects of the digitization project of the journal, including description of standards and metadata can be found in [1] and [2]. The aim of this paper is to discuss several drawbacks in using TEX typesetting system for the retro-digitization that we have encountered recently.

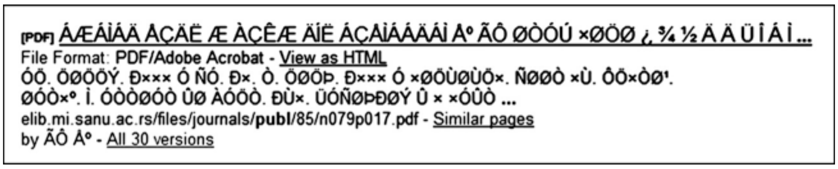
2 Internet Presentation of the Journal

Portable Document Format (PDF) is a file format created by Adobe Systems in 1993 for document exchange. PDF is used for representing two-dimensional documents in a manner independent of the application software, hardware, and operating system. Therefore PDF is a good solution for creating archives for both paper and electronic documents. However, some data standards must be obeyed and metadata included into the document for this purpose, see [11]. Hence, a simple conversion from other document formats to PDF without having in mind these issues my rise certain problems and degrades the use of so obtained documents in PDF format. Here we adduce our experience in this direction concerning Internet presentation of mathematical articles in PDF format published in our journal.

After completing retro-digitization, the Institute decided to make the journal repository freely available via Internet. The editorial board of the journal has understood that besides archiving, the other aim of digitization is to make available journals, books and papers to the general public. Therefore, we prepared Internet presentation of our repository (it can be found at the address http://publications.mi.sanu.ac.rs/) which allowed online browsing and searching according to metadata which includes the names of the authors and the titles of the papers. Downloading of full texts in PDF file format of all journal papers is also possible. When the presentation became fully functional we added it to the Google's index with Google Webmaster Tools [3]. Our aspiration was to take the full advantage of Google PDF files indexing, first of all to increase the visibility and availability of the content of the journal at the Internet. In other words we expected that Google searching engine will find particular words in the bodies of the texts and correlate them to the files containing them. For this reason we designed our presentation so that the PDF file of every journal paper has the unique URL address. In this way we achieved the following:

- All texts in the repository can be fully indexed and searchable via Google, opposed to our local search based on paper's meta-data which is backed up by a database created for that purposes,
- Hyperlinks to journal papers from our repository should appear in the results of search queries performed by Google.

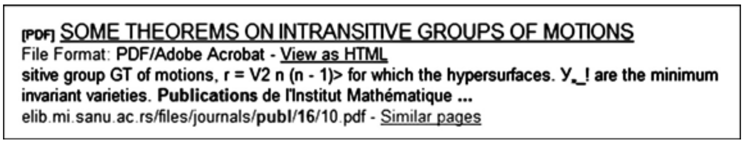
But our surprise was great when we found, almost accidentally, that the result of indexing was not as we had expected it to be. Namely, for certain articles Google returned unrecognizable sequences of characters as can be seen in Figure 1.



[PDF] ÁÆÁÌÁÄ ÁÇÄË Æ ÀÇÊÆ ÄÌË ÁÇÁÌÁÁÄÁÌ Áº ÃÔ ØÒÓÚ ×ØÖØ ¿ ¾ ½ Ä Ä Ü Î Á Ì ...
File Format: PDF/Adobe Acrobat - View as HTML
ÓÖ. ÖØÖÖÝ. Ð××× Ó ÑÓ. Ð×. Ò. ÖØÖÞ. Ð××× Ó ×ØÖÙØÙÖ×. ÑØØÒ ×Ù. ÔÖ×ÒØ¹.
ØÓÒ×º. Ì. ÓÒÒØÓÒ ÛØ ÀÓÖÒ. ÐÙ×. ÜÓÑØÞÐØÝ Û × ×ÓÛÒ ...
elib.mi.sanu.ac.rs/files/journals/**publ**/85/n079p017.pdf - Similar pages
by ÃÔ Áº - All 30 versions

Figure 1: Unrecognizable sequences of characters

Soon we found out that almost all of PDF files generated from $\TeX$ sources suffered from the same problem. On the other side, Google worked well for scanned papers, due to built in OCR in Google's indexing engine as can be seen in Figure 2.



[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

Figure 2: Example of a scanned paper

3 Analysis and Solution

Obviously, such a situation was not acceptable as the main goals were destroyed, the Internet visibility of digital copies and text body searching by Google searching engine. Therefore, we decided to analyze in details the existing PDF files generated from TEX sources and to find out what was going wrong. The final aim of the analysis was the improvement of the file converting process from TEX to PDF format. Many authors practiced similar research for other purposes, for example for extracting mathematical content of PDF documents, see [4].

We analyzed about 1000 files from the repository. Due to the large number of files, the analysis had to be automatic. As the main tool for our analysis we used iText, a Free Java-PDF Library [5]. As a result of the analysis we divided the problematic files into two groups.

Group 1. PDF files with embedded Type 3 fonts, but without ToUnicode CMaps [6]. 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. Namely, the texts in these files consist of separated pictures of the characters in CCITT[1] facsimile (fax) encoding, not of the codes of characters themselves. Files of this type were indexed by Google, due to embedded OCR technology in the Google indexing engine. However, there were a lot of errors, possibly because Google uses some kind of semantics in text recognition. Independently of indexing, we consider that PDF files obtained in this way from TEX sources are unacceptable. For example, zooming texts contained in such files may result in very poor display and printing 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. Data for conversions and software configuration were not stored, so now it is difficult, better to say impossible, to trace them all and say what was wrong in the processes of conversions. 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. It was important how prints appear on the paper, it was not important in what way they were obtained. Also, we would remind that a decade ago, scientific papers were kept and distributed mainly in DVI and PS (PostScript) formats. It seems that very few researches anticipated that new and mighty capabilities on Internet such as the full text indexing would appear soon.

In the recent years, PDF format has become de facto the standard for document presentation and exchange on Internet. According to the remarks we just stated, we consider that it is

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

important to define standards 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:

- Search and find capabilities,
- Copy and paste results,
- Display quality.

Now we shall consider those issues of TEX processing and conversion to PDF format that would lead to output for good distribution and indexing on Internet. 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 isolated text lines. So, for example, the search for phrases and hyphenated words might fail. Additional complication may arise from the way in which the bounds between words are determined, since they are affected not only by separator characters, but by the spacing between characters too. In TEX, and PDF also, 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. However, in PDF files this capability makes the text searching the most difficult task. But we shall 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 the usage by only one application (internal document format for example), problems should not be expected, as the encodings are usually embedded in the application itself. But if PDF files are produced for several, or unknown number of applications, e.g. Internet searching, the Unicode should be used.

So, there are two aspects of the language problem in PDF documents: showing a text and determination of the 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 [7]. In more details, the Unicode standard defines a system for numbering all of the common characters used in a large number of languages. It is a suitable scheme for representing the information content of text, but not its appearance, since Unicode values identify characters, not glyphs. According to PDF Reference [8] a consumer application 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[2].

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 in-

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

dexing. Another advantage of the usage of this relation is the solution of ligatures problem. Namely, it enables a ligature in the selected text to be automatically transformed into separated characters, for example ffi into f f i.

The display quality of the paper is related to the appearance of PDF text in various resolutions, due to screen resolution or zooming. This issue is simply solved by the usage of Type 1 fonts (outline fonts). The Type 1 fonts are scalable in contrast to Type 3 fonts which are bitmapped.

Our concluding remark is that all PDF files intended for multiple usages should have embedded Type 1 fonts with appropriate ToUnicode CMaps. This conclusion leads to the right choice of the software for TEX processing and conversion to PDF format.

The right choice 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. Of course, the choice of proper fonts is also very important. We decided to use Latin Modern fonts which also come together with newly LATEX distributions. At the Internet site CTAN [9] there is a package **cmap** [10], developed by Vladimir 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. An additional reason is that they support most of the font encodings (OT1, T1, T2A, T2B, T2C …). This support is very important for our usage, as the papers in our journal are multilingual and multi-alphabetic.

In order to use the above mentioned techniques the following packages should be included in the preamble of TEX document

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

In fact this is the basic form of the preamble that we are using for generating PDF documents with all the mentioned properties regarding search and find capabilities, copy/paste results and display quality. The suggested techniques are quite appropriate for the future use, but we expect problems in rebuilding PDF files from old TEX sources. This is due to the large amount of papers (about 1000) various styles, macros, the presence in them of several languages and alphabets. It is hardly expected that this job can be done automatically since source TEX files use many packages and styles (AMS-TEX, LATEX, plain TEX etc).

4 TEX and e-books

We shall discuss here the usage of TEX prepared mathematical papers in PDF format by e-book readers. The portability is the other facet concerning texts in the electronic forms. This aspect of digitized documents is particularly interesting in the presence of e-book readers, specialized computers for reading texts in the electronic form. Our experiment with TEX generated documents, especially those containing mathematical formulas, for use with e-book readers is disappointing.

The idea of making e-books mobile rather than just viewable on a computer has been as inevitable as promising. Therefore, it is no surprise that the current e-book reader market appears to be a highly competitive environment with a somewhat open outcome of who is going to be the users' favourite. Springer Verlag already offers 30 000 titles of e-books, and the whole production since 2005 appears in electronic form too. The US market has already seen the introduction of both Amazon's Kindle and the Sony Reader, whereas users in Europe are still awaiting (except in the United Kingdom) a more coherent market offering.

We experimented with Sony e-book reader PRS-505 [15]. Besides very good portability, it has exceptionally fine display (170 dpi) comparable to the paper print quality. This is due to the new technologies, e-paper [16] and e-ink [17]. Simple typed text, such as fiction, particularly prepared by Sony Corporation looks fine on it too. But all advantages of this apparatus stop here, at least if it concerns reading mathematical texts prepared in TEX. We have randomly chosen several mathematically oriented papers, converted them to PDF and used them on the e-book reader. The first appearance of the text on the screen was fine. But in most cases characters were too small for reading, and we wanted to use zooming capability of the device, which otherwise works fine for e-books which comes with it. The disaster starts at that point. Let us mention just few of the problems. Text formatting was destroyed, some formulas were wrongly displayed, even overlapped, tables were destroyed and some pages could not be zoomed, hence they were unreadable. Even in default text size, the supplied software did not obey embedded formatting commands, such as cropping. When we prepared mathematical texts specifically for this device, they looked good at default size but the problems remain if the texts are zoomed. Unfortunately, we usually want to read a text that comes from other authors which we cannot reformat, and in most cases this device is useless.

At the first glance, the issue seems to be primarily one of the poor font-size choices and/or zooming capabilities of the device itself and its incorporated PDF software, rather than of the PDF file format itself, or how it was generated. A deeper insight shows that the deficiencies lay in the supplied Acrobat Reader and some limitations of the existing TEX processing system. Namely, the software that comes with Sony Reader is based on *Adobe Digital Editions* [12]. This software is a rich Internet application (RIA) built exclusively for digital publishing. It is focused and lightweight solution for a simplified, engaging way to acquire, manage, and read e-books. It supports PDF/A-1 standard, a file format for the long-term archiving of electronic documents.

PDF/A standard [14] is based on the PDF Reference Version 1.4 from Adobe Systems Inc. (implemented in Adobe Acrobat 5) and is defined by ISO 19005-1:2005. It identifies a structure for electronic documents that ensures the documents can be reproduced in the exactly same way in the future. A key element to this reproducibility is the requirement for PDF/A documents to be completely self-contained. Therefore, all of the information necessary for displaying the document in the same manner every time is embedded in the file. This includes all content, such as text, raster images and vector graphics, fonts, and colour information. A PDF/A document are not permitted to be reliant on information from external sources (e.g. font programmes). Also, JavaScript and executable file launches are prohibited and all fonts must be embedded and also must be legally embeddable for unlimited, universal rendering. This also applies to the PostScript standard fonts such as Times or Helvetica. Colour spaces specified in a device-independent manner and encryption is disallowed. Finally, the usage of standards-based metadata is mandated. PDF/A-1 is currently used version, while PDF/A-2 was a new part to the standard, ISO 19005-1, Part-2 (PDF/A-2), is currently being elaborating only by the Technical Committee. PDF/A-1 has two conformance levels, PDF/A-1a and PDF/A-1b. PDF/A-1b has the objective of ensuring reliable reproduction of the visual appearance of the document, while PDF/A-1a includes all the requirements of PDF/A-1b and additionally requires that the document structure to be included. Therefore, the text must be tagged, and should include ToUnicode CMaps, with the objective of ensuring that the document content can be searched and repurposed, such as for reformatting, particularly reflowing for display on small size screens (Sony Reader has a rather small screen, only 15cm in diagonal).

Sony Reader in zooming mode is doing in fact the reorganization (re-flow) of the text, and for this purpose the text must be tagged. On the other hand, the current version of pdfLaTeX does not support text tagging[3], and this is the main and insuperable drawback in producing PDF files by the TEX processing system of mathematical texts, or any texts with complex structures, such as having tables or any "two-dimensional" constructs such as the matrices and the complex mathematical formulas.

5 Conclusion

We used TEX in the nineties for the retro-digitization of our journal *Publications de l'Institut Mathématique*. Simply we retyped in TEX all volumes of the journal printed since 1980. When converted in PDF and displayed on Internet, Google indexed them erroneously. This was mainly due to an inconsistent and careless conversion from TEX to PDF. We propose a solution for conversion of TEX generated texts to PDF. The main components are the usage of Type 1 fonts (due to scalability) and ToUnicode CMap relation, needed for determination of text content. Portability is the other facet concerning texts in the electronic forms. This aspect of digitized documents is particularly interesting in the presence of e-book readers, specialized computers for reading texts in the electronic form. Our experiment with TEX generated documents, especially those containing mathematical formulas, for use with these devices with current versions is disappointing, mainly due to some limitations of producing PDF files with pdfLaTeX.

References

1 Ž. Mijajlović, Z. Ognjanović, *Digitization of Mathematical Editions in Serbia*, Proc. DML 2008, ed. Petr Sojka, Birmingham, UK, July 27th, 2008, Masaryk Univ. Brno 2008, 87-96.

2 Ž. Mijajlović, Z. Ognjanović, A. Pejović, *Internet presentations of mathematical works in Serbia*, NCD Review 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 iText, a Free Java-PDF Library, http://www.lowagie.com/iText/.

6 Adobe CMap and CIDFont Files Specification, http://www.adobe.com/devnet/font/pdfs-/5014.CIDFont_Spec.pdf.

7 ToUnicode Mapping File Tutorial, http://www.adobe.com/devnet/acrobat/pdfs/5411.ToUnicode.pdf.

8 PDF Reference 1.7, 6th Ed., http://www.adobe.com/devnet/pdf/.

9 The Comprehensive TEX Archive Network, http://www.ctan.org/.

10 The cmap package developed by Vladimir Volovich can be obtained from http://tug.ctan.org/tex-archive/macros/latex/contrib/cmap/.

11 PDF as a standard for archiving, white paper, http://www.adobe.com/enterprise/pdfs/pdfarchiving.pdf.

12 Adobe Digital Editions, http://www.adobe.com/products/digitaleditions/.

13 Patch to experiment with tagged PDF, developed by Hàn Thế Thành, can be obtained from http://sarovar.org/tracker/index.php?func=detail&aid=945.

[3] Actually there is a patch to experiment with tagged PDF [13]. It is developed by Hàn Thế Thành, but it cannot be considered as an out-of-the-box solution since it is still in very alpha stage.

14 PDF/A Wikipedia page, http://en.wikipedia.org/wiki/PDF/A, PDF/A Competence Center, http://www.pdfa.org/.
15 Sony e-book reader PRS-505 features/specifications, http://www.sony.co.uk/product/rd-reader-ebook-/prs-505.
16 Electronic paper, http://en.wikipedia.org/wiki/Electronic_paper.
17 E-Ink: Electronic paper displays, http://en.wikipedia.org/wiki/E_Ink.

pejovica@mi.sanu.ac.rs
zarkom@matf.bg.ac.rs