Creating high-quality PDF/A documents using LaTeX

This document provides step-by-step instructions for generating valid PDF/A from LaTeX sources. (Written 2014/08/17. Revised 2016/11/13).

This page is still a draft. The files and instructions below are still subject to change, even in ways that aren't backward compatible. If you follow these instructions and find any mistakes, or run into problems you cannot solve, please feel free to email me at to selinger@mathstat.dal.ca. I will do my best to improve this document and add troubleshooting hints where necessary.

Contents

1. A brief introduction to PDF/A

A PDF/A document is a special kind of PDF document that has been optimized for long-term archiving. PDF/A is a standard of the International Organization for Standardization (ISO). Some of the main features of PDF/A documents are:

- **Self-containedness:** this means that all resources that are required to reproduce the document's visual appearance, such as fonts, color spaces, etc., are embedded within the document itself. In particular, the document must be device independent.
- **Unicode mapping:** this means that all of the document's content has been mapped to machine readable Unicode text. Such a mapping makes the document searchable, allows text to be copied and pasted, and allows text to be displayed in other forms (such as via a screen reader for the blind).
- **Metadata:** PDF/A specifies a standardized format for including metadata, such as the document's title, author, copyright, keywords, abstract, publication data, and so on. This helps to ensure that the document can be found and correctly indexed by search engines, libraries, etc.

There are various different versions and levels of PDF/A, differing in the features that are supported and the strictness of the compliance requirements. For the purpose of this document, we will mainly be concerned with the PDF/A-1b standard, which mandates self-containedness and the presence of metadata.

We will also be concerned with Unicode mapping, which is optional, but highly desirable, in PDF/A-1b documents.

2. PDF/A validation

Many organizations, such as universities, require certain documents, such as Master's and Ph.D. theses, to be prepared in PDF/A format. This makes sense, because such documents will be archived, and should remain readable for the indefinite future.

In such situations, either the document's author, or the institution, or both, will perform a "check" to ensure that the document actually conforms to the PDF/A requirements. When it is done correctly, this check has two parts: automated validation and manual validation.

- **Automated validation** can be performed by software. Its purpose is to check that the document conforms to the *technical requirements* of the PDF/A standard (for example, presence of metadata, presence of Unicode mapping, presence of embedded fonts, etc).
- **Manual validation** must be performed by a person. Its purpose is to check that the document conforms to the *intent* of the PDF/A standard (for example, correctness and appropriateness of the metadata, correctness of the Unicode mapping).

There are a number of software tools available that perform automated validation of PDF/A documents. Probably the most common of these is the "Preflight" tool in Adobe® Acrobat® Pro ("View" → "Tools" → "Print Production" → "Preflight"). Unfortunately, Acrobat Pro is not free software. There are also a number of free and online tools available. Just google them. Note that not all tools are equivalent: a document validated by one tool could still fail validation using another.

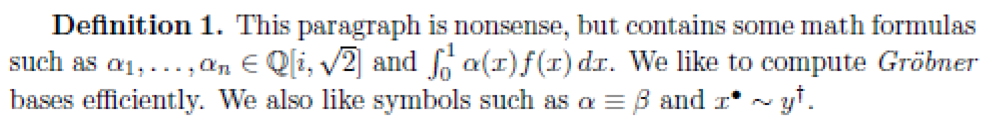
3. Generating PDF/A from LaTeX: How not to do it

Some institutions cut corners by relying exclusively on software validation of PDF/A documents, skipping manual validation. This encourages authors to submit PDF/A documents of *poor quality*, i.e., documents that meet the technical requirements, but not the intent, of the PDF/A standard.

Almost any PDF document can easily be converted to PDF/A-1b, using automated software tools such as the "Convert to PDF/A-1b" option of the Preflight tool of Acrobat Pro. So you could generate a "plain old" PDF file from your LaTeX sources, and then convert it to PDF/A using Acrobat Pro.

While one can sometimes get away with this method, the problem is that the resulting PDF/A-1b is of poor quality: if the input document did not contain metadata or Unicode mappings, then the conversion software will not be able to create this data out of thin air. It will typically generate empty metadata and incorrect Unicode mappings.

For example, consider two PDF/A documents containing this paragraph of text:



Definition 1. This paragraph is nonsense, but contains some math formulas such as $\alpha_1, \ldots, \alpha_n \in \mathbb{Q}[i, \sqrt{2}]$ and $\int_0^1 \alpha(x) f(x)\,dx$. We like to compute *Gröbner* bases efficiently. We also like symbols such as $\alpha \equiv \beta$ and $x^\bullet \sim y^\dagger$.

The first document example1.pdf was generated from LaTeX sources using `pdflatex` without any special steps, and then converted to PDF/A-1b using Acrobat Pro. Let us try copying and pasting its contents into a text file. The result is gibberish:

```
Denition 1. This paragraph is nonsense, but contains some math
formulas such as 1; : : : ; n 2 Q[i; p 2] and R 1 0 (x)f(x) dx.
We like to compute Gobner bases eciently. We also like symbols such
as    and x  yy.
```

It is obvious that the document will not be searchable, nor can it be read aloud by a screen reader. Foreign letters such as "ö" and "α", ligatures such as "fi" and "ffi", and most mathematical symbols have simply been omitted, while some other symbols were translated to random characters.

The second document example2.pdf was generated by the methods described below. When copying and pasting from this document, we get this result:

```
Definition 1. This paragraph is nonsense, but contains some math
formulas such as α1, . . . , αn ∈ Q[i, √ 2] and ∫ 1 0 α(x)f(x) dx.
We like to compute Gr¨obner bases efficiently. We also like symbols
such as α ≡ β and x• ~ y†.
```

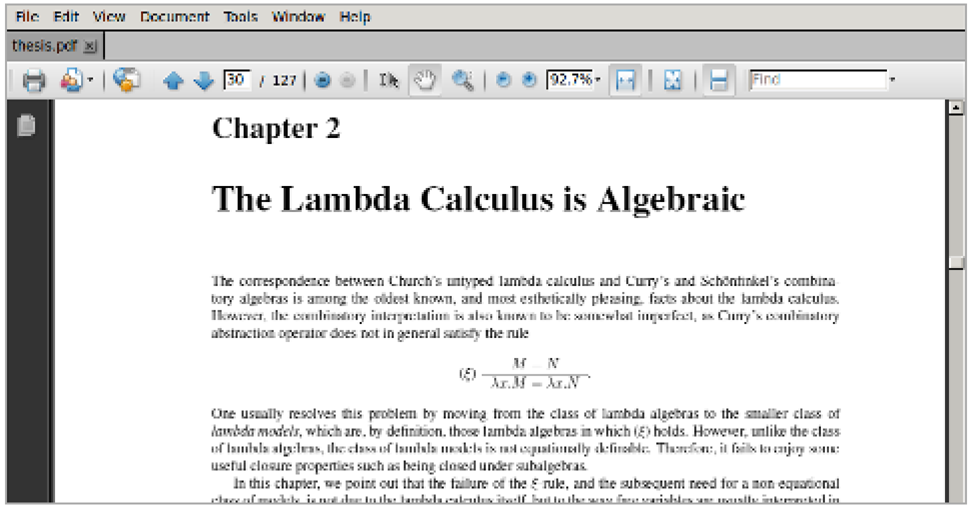
While this is not absolutely perfect (for example, there are no subscripts and superscripts), it is certainly quite readable, and much better than the above.

4. Generating PDF/A from LaTeX: Step-by-step instructions

Here are my step-by-step instructions for converting an existing LaTeX document to PDF/A. These steps should work for most LaTeX documents.

To illustrate the steps, I will use my Ph.D. thesis as a running example. I wrote my thesis in 1997, long before I had ever heard of PDF, let alone PDF/A.

Initial version: thesis.tex, thesis.pdf



Chapter 2

The Lambda Calculus is Algebraic

The correspondence between Church's untyped lambda calculus and Curry's and Schönfinkel's combinatory algebras is among the oldest known, and most esthetically pleasing, facts about the lambda calculus. However, the combinatory interpretation is also known to be somewhat imperfect, as Curry's combinatory abstraction operator does not in general satisfy the rule

$$(\xi)\ \frac{M = N}{\lambda x.M = \lambda x.N}.$$

One usually resolves this problem by moving from the class of lambda algebras to the smaller class of *lambda models*, which are, by definition, those lambda algebras in which (ξ) holds. However, unlike the class of lambda algebras, the class of lambda models is not equationally definable. Therefore, it fails to enjoy some useful closure properties such as being closed under subalgebras.

In this chapter, we point out that the failure of the ξ rule, and the subsequent need for a non-equational

1. **Prerequisites:** you must have `pdflatex` installed, as well as the `hyperref` and `xmpincl` packages. Most modern TeX distributions have these things preinstalled.

   Option 1: If you want to generate PDF/A compliant documents "out of the box", you need at least pdfTeX 1.40.15, and preferably pdfTeX 1.40.16. As of this writing in August 2014, this version is not yet included in current software distributions such as Ubuntu, but it is available for installation from Tex Live. You can type `pdflatex --version` to determine your version.

   Option 2: Even if you have an older version of pdfTeX (such as 1.40.14, which I am currently using), you can still generate PDF/A compliant documents if you also have access to Acrobat Pro. This requires two steps: first use `pdflatex` to generate an almost PDF/A compliant document, and then use Acrobat Pro to repair the few remaining errors.

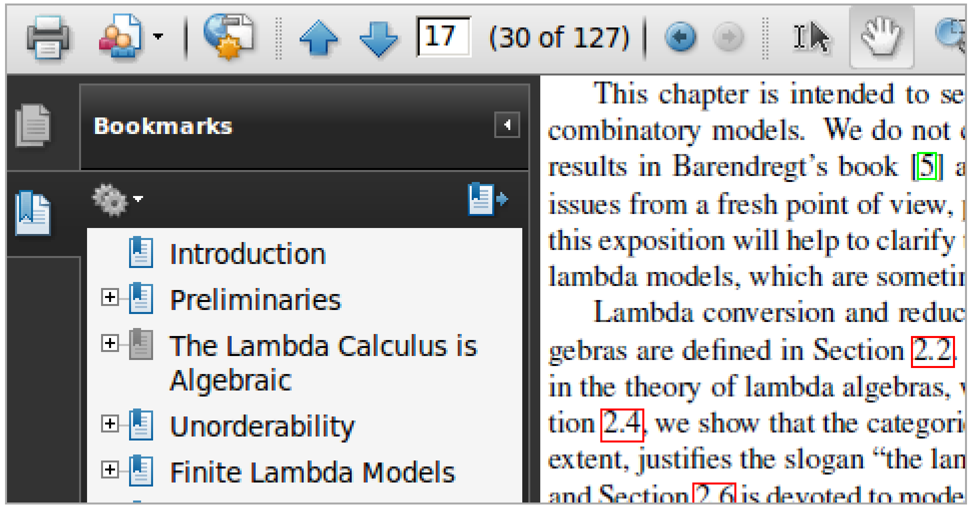
2. **Find your main source file.** I will assume, for the sake of argument, that your main LaTeX file is called `thesis.tex`. If it has a different name, adjust the following instructions accordingly.

3. **Activate hyperref.** Add `\usepackage{hyperref}` to the preamble of your LaTeX source. The preamble is the part of the file between `\documentclass` and `\begin{document}`. Do not specify any options to the hyperref package. Then try compiling it with

   ```
   pdflatex thesis.tex
   ```

   You may get some errors the first time around, but it should compile correctly the second time. If everything goes well, look at the resulting PDF file in Acrobat Reader. (You may have to select "File" → "Reload" if the file was already open before). You should notice a few changes: cross-references (theorems, equations, citations, page numbers, etc.) now show up as hyperlinks, and there should be a sidebar of PDF bookmarks. There are also other subtle changes; for example, Acrobat Reader now correctly understands that "page 17" is actually the 30th page of my thesis.

   With hyperref: thesis-step3.tex, thesis-step3.pdf



This chapter is intended to se
combinatory models. We do not
results in Barendregt's book [5] a
issues from a fresh point of view,
this exposition will help to clarify
lambda models, which are sometir
Lambda conversion and reduc
gebras are defined in Section 2.2
in the theory of lambda algebras,
tion 2.4, we show that the categori
extent, justifies the slogan "the lan
and Section 2.6 is devoted to mode

If anything goes wrong during this step, I can't offer much help. Try the documentation of the hyperref package, or try googling the error message.

4. **Set up PDF/A files.** Make sure that you have at least version 1.5.8 of the pdfx package. This is probably preinstalled if your TeX distribution is from 2016 or newer. Please note that older versions of the pdfx package are buggy and do not work. If you don't have at least version 1.5.8, get it from CTAN:

   - The pdfx package at CTAN.

   Copy the following file to the directory where your LaTeX sources are (your "source directory"). It may be helpful to right-click on the link and select "Save Link As".

   - sample.xmpdata

5. **Rename sample.xmpdata.** Rename the `sample.xmpdata` file as `thesis.xmpdata`, of course using the actual base name of your LaTeX source file instead of `thesis`.

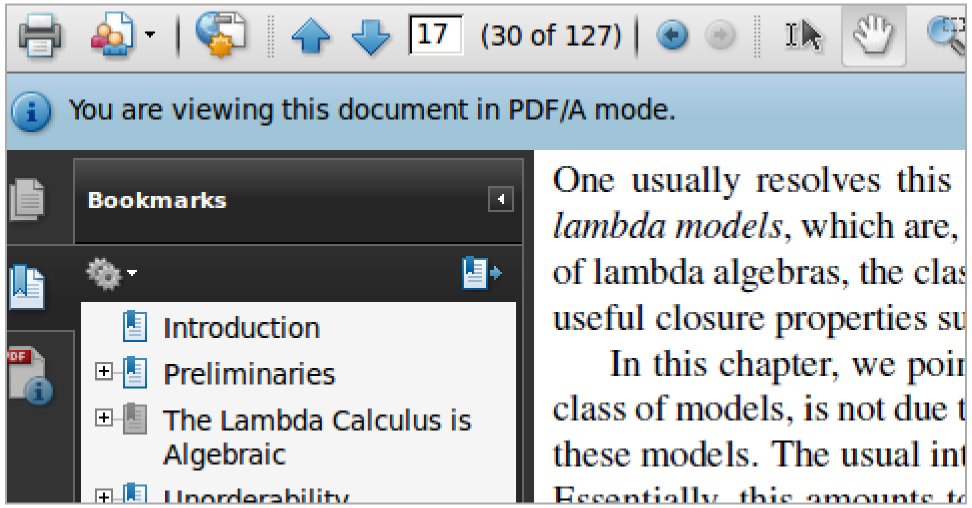
6. **Activate pdfx.** Add `\usepackage[a-1b]{pdfx}` to the preamble of your LaTeX source. Note that you must put this *before* the `\usepackage{hyperref}`.Then try compiling it with

   ```
   pdflatex thesis.tex
   ```

   In this step, I received one error, because my document defined the macro `\B`, which was internally used by the pdfx package. I changed `\newcommand{\B}{B}` to `\renewcommand{\B}{B}` and this fixed the problem.

   Assuming there are no further errors, open the resulting PDF file in Acrobat Reader. (You may have to select "File" → "Reload" if the file was already open before). Acrobat Reader should now show a blue header announcing that "You are viewing this document in PDF/A mode." This does not yet mean that your file has been validated to be valid PDF/A. It only means that the file claims to be PDF/A.

With pdfx: [thesis-step6.tex](), [thesis-step6.xmpdata](), [thesis-step6.pdf]()

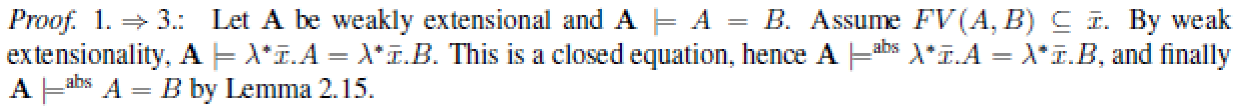


You are viewing this document in PDF/A mode.

One usually resolves this
lambda models, which are,
of lambda algebras, the clas
useful closure properties su
In this chapter, we poi
class of models, is not due t
these models. The usual in

7. **Check for Unicode mapping.** At this point, you should be able to copy and paste some text from your PDF/A file and get reasonable output. Try copying and pasting some math formulas, text containing ligatures, etc. Of course, one cannot expect every complex math formula to be converted to sensible Unicode; however, you should see a marked improvement in simple mathematical symbols, Greek letters, and so on.

   For example, copying and pasting the beginning of the Proof of Proposition 2.22 from page 27

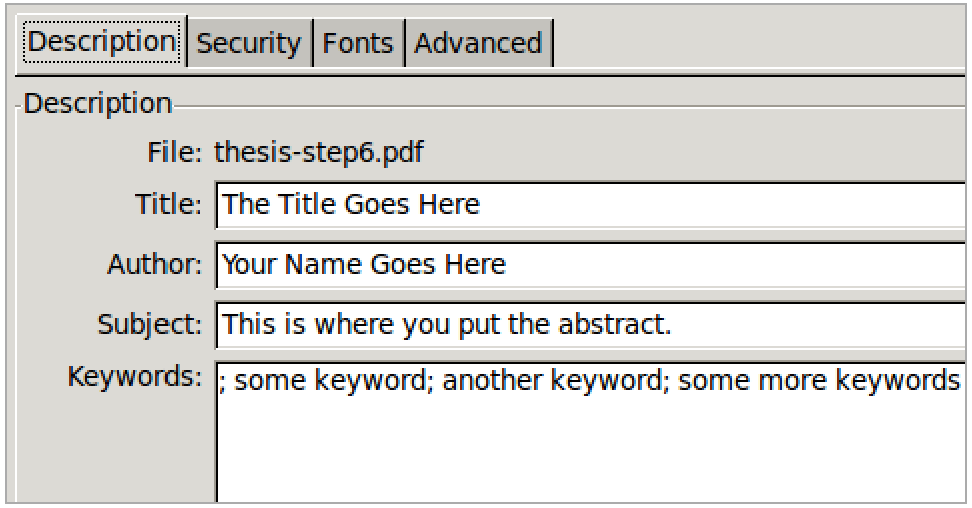
   

   Proof. 1. $\Rightarrow$ 3.: Let $\mathbf{A}$ be weakly extensional and $\mathbf{A} \models A = B$. Assume $FV(A,B) \subseteq \bar{x}$. By weak extensionality, $\mathbf{A} \models \lambda^*\bar{x}.A = \lambda^*\bar{x}.B$. This is a closed equation, hence $\mathbf{A} \models^{\text{abs}} \lambda^*\bar{x}.A = \lambda^*\bar{x}.B$, and finally $\mathbf{A} \models^{\text{abs}} A = B$ by Lemma 2.15.

   gives the following result:

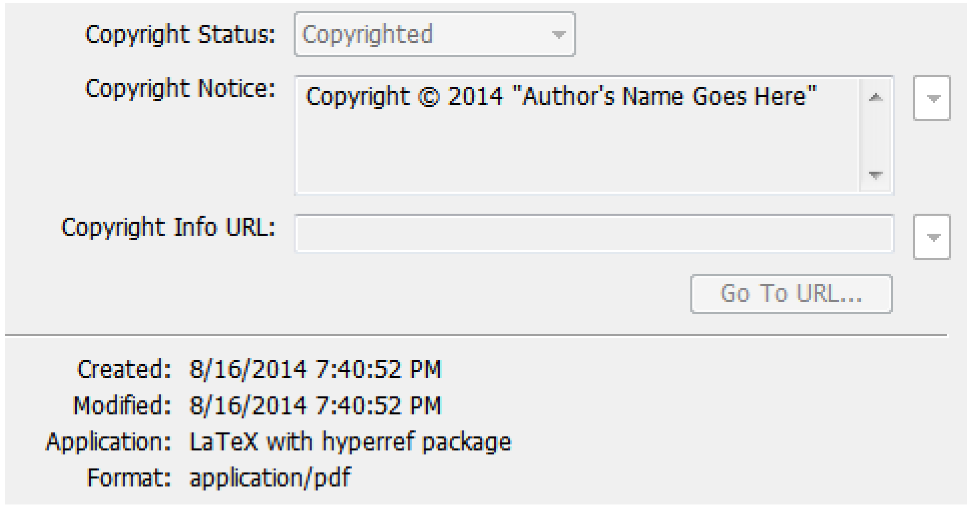
```
Proof. 1. ⇒ 3.: Let A be weakly extensional and A |= A = B. Assume FV (A,B) ⊆ ¯x.
By weak extensionality, A |= λ∗¯x.A = λ∗¯x.B. This is a closed equation, hence
A |=abs λ∗¯x.A = λ∗¯x.B, and finally A |=abs A = B by Lemma 2.15.
```

8. **Check for metadata.** In Acrobat Reader, go to "File" → "Properties". You should see a sample title, sample author, sample subject, and sample keywords.



File: thesis-step6.pdf
Title: The Title Goes Here
Author: Your Name Goes Here
Subject: This is where you put the abstract.
Keywords: ; some keyword; another keyword; some more keywords

If you have Acrobat Pro, you can also click on "Additional Metadata", and you should also see a sample copyright notice.



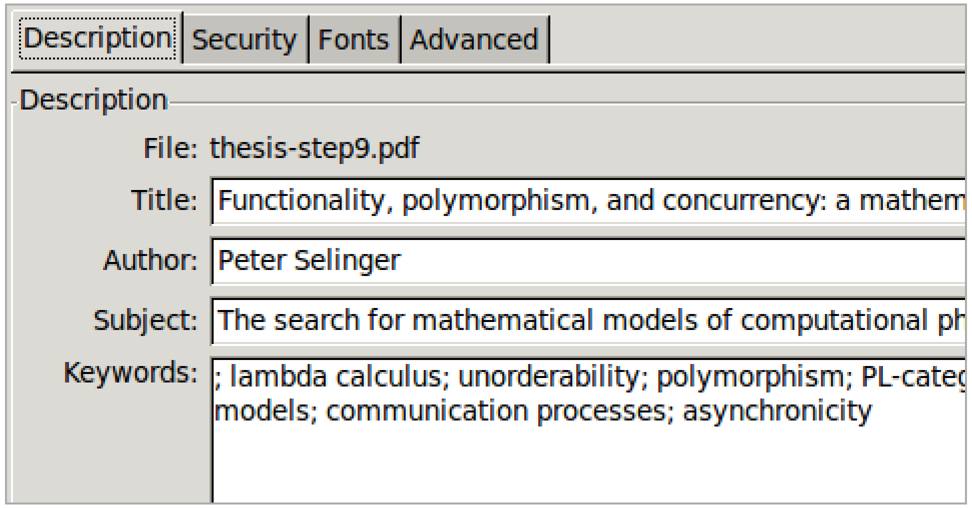
Copyright Status: Copyrighted
Copyright Notice: Copyright © 2014 "Author's Name Goes Here"
Copyright Info URL:

Created: 8/16/2014 7:40:52 PM
Modified: 8/16/2014 7:40:52 PM
Application: LaTeX with hyperref package
Format: application/pdf

9. **Edit the metadata.** Now you should edit the file `thesis.xmpdata` to create the metadata that is appropriate for your document. You can delete any entry (such as `\Keywords` or `\Subject`) that you do not need, and edit the remaining ones to suit your purpose. You can use UTF-8 encoded Unicode in this file, in case you would like to include any foreign letters, math formulas, etc. See Section 5, "Metadata format", for more details on what you can put here.

   Then recompile your document, reopen the file in Acrobat, and check that your metadata shows up correctly. (Don't forget "File" → "Reload" in case the file is still open from before).

With metadata: thesis-step9.tex, thesis-step9.xmpdata, thesis-step9.pdf


Description Security Fonts Advanced

Description

File: thesis-step9.pdf

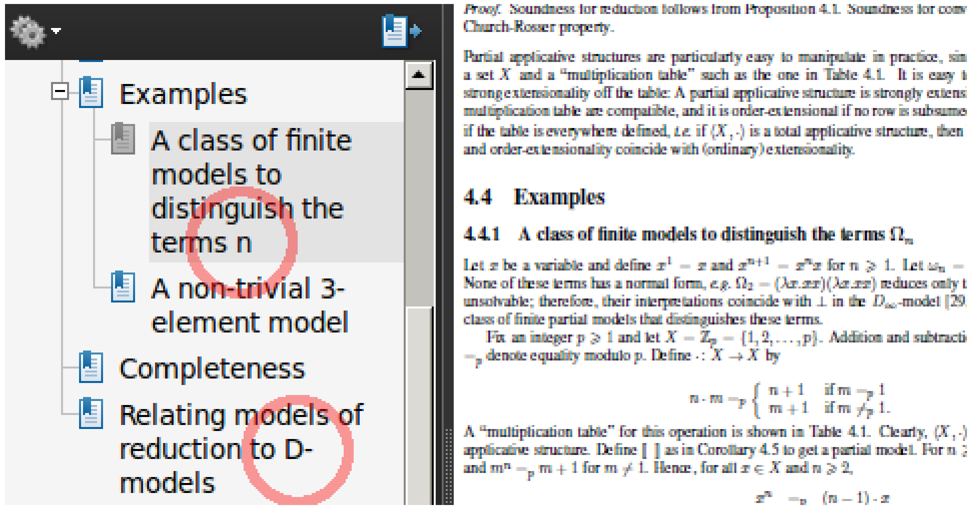
Title: Functionality, polymorphism, and concurrency: a mathem

Author: Peter Selinger

Subject: The search for mathematical models of computational ph

Keywords: ; lambda calculus; unorderability; polymorphism; PL-categ
models; communication processes; asynchronicity

10. **Fix the PDF bookmarks.** When you open your document in Acrobat Reader or Acrobat Pro, you should see a bookmarks sidebar. It should roughly correspond to the table of contents of your document. Check all of the entries. If any of the entries contain math formulas, they will probably look messed-up. For example, my thesis contains a section called "*A class of finite models to distinguish the terms* Ω_n", but it may show up in the PDF bookmarks as "*A class of finite models to distinguish the terms n*". Also, instead of "D_∞*-models*" we may see "*D-models*".


Examples

A class of finite models to distinguish the terms n

A non-trivial 3-element model

Completeness

Relating models of reduction to D-models

Proof. Soundness for reduction follows from Proposition 4.1. Soundness for conv
Church-Rosser property.

Partial applicative structures are particularly easy to manipulate in practice, sin
a set X and a "multiplication table" such as the one in Table 4.1. It is easy t
strong extensionality off the table: A partial applicative structure is strongly extens
multiplication table are compatible, and it is order-extensional if no row is subsume
if the table is everywhere defined, *i.e.* if $(X, \cdot)$ is a total applicative structure, then
and order-extensionality coincide with (ordinary) extensionality.

4.4 Examples

4.4.1 A class of finite models to distinguish the terms Ω_n

Let x be a variable and define $x^1 = x$ and $x^{n+1} = x^n x$ for $n \geqslant 1$. Let $\omega_n =$
None of these terms has a normal form, *e.g.* $\Omega_2 = (\lambda x.xx)(\lambda x.xx)$ reduces only t
unsolvable; therefore, their interpretations coincide with $\bot$ in the D_∞-model [29.
class of finite partial models that distinguishes these terms.

Fix an integer $p \geqslant 1$ and let $X = \mathbb{Z}_p = \{1, 2, \dots, p\}$. Addition and subtracti
$=_p$ denote equality modulo p. Define $\cdot : X \to X$ by

$$n \cdot m =_p \begin{cases} n+1 & \text{if } m =_p 1 \\ m+1 & \text{if } m \not=_p 1. \end{cases}$$

A "multiplication table" for this operation is shown in Table 4.1. Clearly, $(X, \cdot)$
applicative structure. Define $[\![\ ]\!]$ as in Corollary 4.5 to get a partial model. For n ;
and $m^n =_p m+1$ for $m \neq 1$. Hence, for all $x \in X$ and $n \geqslant 2$,

$$x^n =_p (n-1) \cdot x$$

More complicated math formulas, particularly those involving `\boldmath` or similar commands, can lead to even-stranger-looking garbage characters in the bookmarks. There are two options for fixing this:

Option 1: Using plain ASCII.

We will specify an alternate plain ASCII version of each math formula in a section title, using the command `\texorpdfstring`. For example, replace

```
\section{A class of finite models to distinguish the terms $\Omega_n$}
```

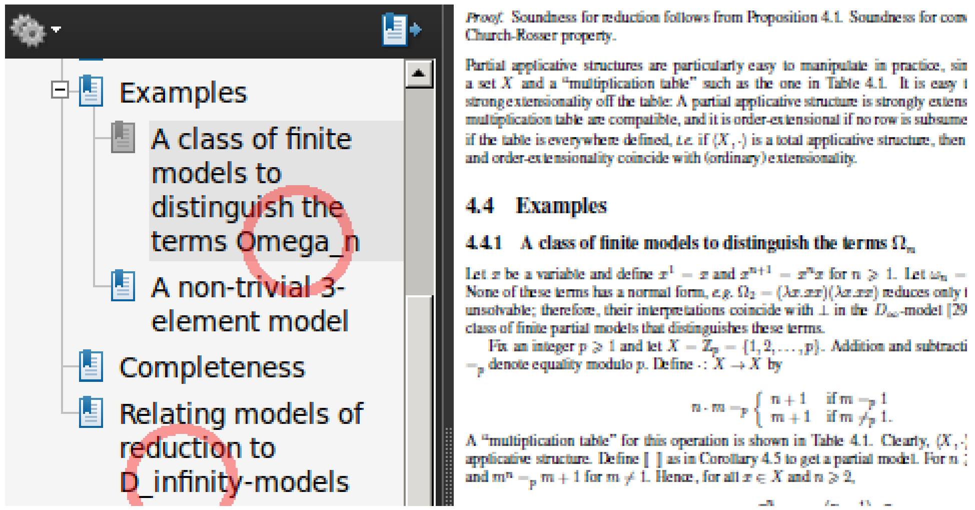
by

```
\section{A class of finite models to distinguish the terms \texorpdfstring{$\Omega_n$}{Omega\_n}}
```

The `\texorpdfstring` command is very simple. It takes two arguments. The first one is what will be typeset in the document, and the second one is the ASCII equivalent that will go into the bookmarks section. Note that the ASCII you can use here is very basic: for example, you can't specify superscripts or subscripts, and if you want to use an underscore "_", you have to escape it as "_". Some other symbols, such as "^", can't be used at all.

Note that you have to run `pdflatex` at least twice for any changes to take effect. Here is what the output looks like:

ASCII bookmarks: thesis-step10a.tex, thesis-step10a.xmpdata, thesis-step10a.pdf


Examples
A class of finite models to distinguish the terms Omega_n
A non-trivial 3-element model
Completeness
Relating models of reduction to D_infinity-models

Proof. Soundness for reduction follows from Proposition 4.1. Soundness for com
Church-Rosser property.

Partial applicative structures are particularly easy to manipulate in practice, sin
a set X and a "multiplication table" such as the one in Table 4.1. It is easy t
strong extensionality off the table: A partial applicative structure is strongly extens
multiplication table are compatible, and it is order-extensional if no row is subsume
if the table is everywhere defined, *i.e.* if $(X, \cdot)$ is a total applicative structure, then
and order-extensionality coincide with (ordinary) extensionality.

4.4 Examples

4.4.1 A class of finite models to distinguish the terms Ω_n

Let x be a variable and define $x^1 = x$ and $x^{n+1} = x^n x$ for $n \geqslant 1$. Let $\omega_n =$
None of these terms has a normal form, *e.g.* $\Omega_2 = (\lambda x.xx)(\lambda x.xx)$ reduces only t
unsolvable; therefore, their interpretations coincide with $\perp$ in the D_∞-model [29
class of finite partial models that distinguishes these terms.

Fix an integer $p \geqslant 1$ and let $X = \mathbb{Z}_p = \{1, 2, \ldots, p\}$. Addition and subtracti
$=_p$ denote equality modulo p. Define $\cdot : X \to X$ by

$$n \cdot m =_p \begin{cases} n+1 & \text{if } m =_p 1 \\ m+1 & \text{if } m \neq_p 1. \end{cases}$$

A "multiplication table" for this operation is shown in Table 4.1. Clearly, $(X, \cdot$
applicative structure. Define $[\![\ ]\!]$ as in Corollary 4.5 to get a partial model. For n
and $m^n =_p m+1$ for $m \neq 1$. Hence, for all $x \in X$ and $n \geqslant 2$,

Option 2: Using Unicode.

To use Unicode characters in the PDF bookmarks, place the following four commands in your LaTeX preamble:

```
\usepackage{inputenc}
\hypersetup{pdfencoding=unicode}
```

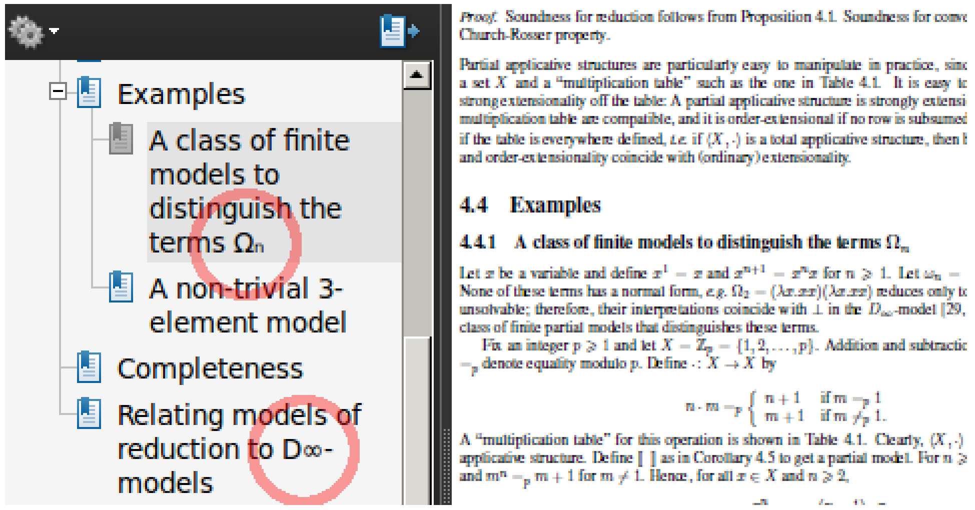
```
\inputencoding{utf8}
\makeatother
```

Note that this will change the input encoding of your LaTeX source to UTF-8. It will also allow you to use Unicode characters in the second argument of `\texorpdfstring`. For example:

```
\section{A class of finite models to distinguish the terms \texorpdfstring{$\Omega_n$}{Ωₙ}}
```

The result looks rather nice:

Unicode bookmarks: thesis-step10b.tex, thesis-step10b.xmpdata, thesis-step10b.pdf


Examples
A class of finite models to distinguish the terms Ωₙ
A non-trivial 3-element model
Completeness
Relating models of reduction to D∞-models

Proof. Soundness for reduction follows from Proposition 4.1. Soundness for conve
Church-Rosser property.

Partial applicative structures are particularly easy to manipulate in practice, sinc
a set X and a "multiplication table" such as the one in Table 4.1. It is easy to
strong extensionality off the table: A partial applicative structure is strongly extensi
multiplication table are compatible, and it is order-extensional if no row is subsumed
if the table is everywhere defined, *i.e.* if $(X, \cdot)$ is a total applicative structure, then b
and order-extensionality coincide with (ordinary) extensionality.

4.4 Examples

4.4.1 A class of finite models to distinguish the terms Ω_n

Let x be a variable and define $x^1 = x$ and $x^{n+1} = x^n x$ for $n \geqslant 1$. Let $\omega_n =$
None of these terms has a normal form, *e.g.* $\Omega_2 = (\lambda x.xx)(\lambda x.xx)$ reduces only to
unsolvable; therefore, their interpretations coincide with $\perp$ in the D_∞-model [29,
class of finite partial models that distinguishes these terms.

Fix an integer $p \geqslant 1$ and let $X = \mathbb{Z}_p = \{1, 2, \ldots, p\}$. Addition and subtractio
$=_p$ denote equality modulo p. Define $\cdot : X \to X$ by

$$n \cdot m =_p \begin{cases} n+1 & \text{if } m =_p 1 \\ m+1 & \text{if } m \neq_p 1. \end{cases}$$

A "multiplication table" for this operation is shown in Table 4.1. Clearly, $(X, \cdot)$
applicative structure. Define $[\![\ ]\!]$ as in Corollary 4.5 to get a partial model. For $n \geqslant$
and $m^n =_p m + 1$ for $m \neq 1$. Hence, for all $x \in X$ and $n \geqslant 2$,

Of course, here I have presupposed that you know how to enter Unicode characters in the editor that you use to write your LaTeX file. I usually do this by copying and pasting the character from some website. For example, if you google "Unicode subscript n", you'll find a browser test page for that character, from which you can then copy and paste.

If you don't want to enter actual Unicode characters in your LaTeX file, TeX also offers a way to escape them. This requires you to look up the UTF-8 encoding of each character, which you can also find via Google (scroll down to "UTF-8 (hex)"). For example, the UTF-8 encoding of "Omega" is `0xCE 0xA9`, and the UTF-8 encoding of "subscript n" is `0xE2 0x82 0x99`. You can then enter the characters in your LaTeX file like this:

```
\section{A class of finite models to distinguish the terms \texorpdfstring{$\Omega_n$}{^^ce^^a9^^e2^^82^^99}}
```

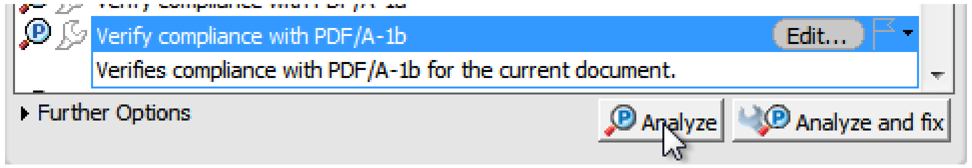
An example file is thesis-step10c.tex; the output is identical to the above.

11. **Verify PDF/A-1b compliance.** If your pdfTeX version is at least 1.40.15 (see Step 1: Prerequisites), then your generated document *should* already comply with the PDF/A-1b standard at this point.

However, it is possible, for a variety of reasons, that you did everything correctly, and your document still doesn't comply. For example, PDF/A-1b does not permit transparency in images, so if you included any images that use transparency, your document may not be compliant. Or you may have included an image that uses the wrong color space. Some font that you used may have been slightly buggy. Or you may have used an older version of pdfTeX.

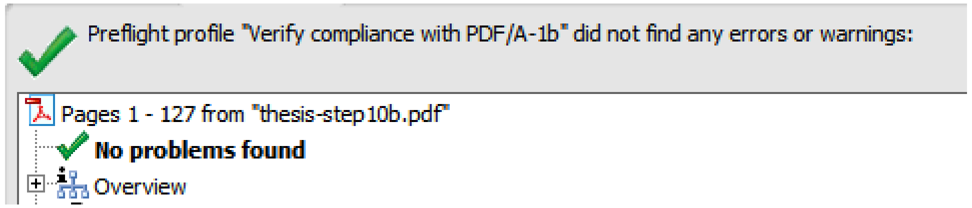
Don't worry. In Steps 11 and 12, we will use Acrobat Pro to verify that our document complies with the PDF/A-1b standard, or to fix it if necessary. If you don't have access to Acrobat Pro, you may also try one of the other tools mentioned in Section 2 above.

To verify compliance in Acrobat Pro, go to "View" → "Tools" → "Print Production" → "Preflight" → "Verify compliance with PDF/A-1b" → "Analyze".


Verify compliance with PDF/A-1b

Verifies compliance with PDF/A-1b for the current document.

If your document passes the test, you get a report of "No problems found".


Preflight profile "Verify compliance with PDF/A-1b" did not find any errors or warnings:

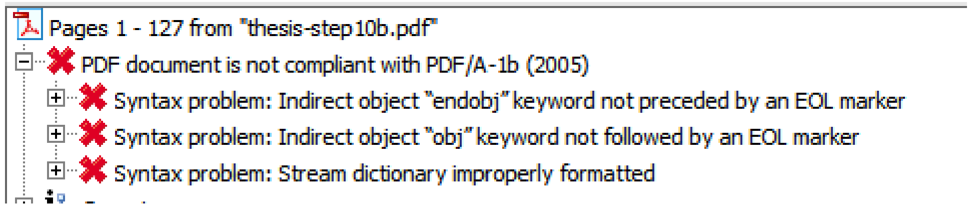
Pages 1 - 127 from "thesis-step10b.pdf"

No problems found

Overview

In this case, you are finished, and you can skip Step 12. However, if you get some errors here, go to Step 12 to fix them.

12. **If necessary, convert to PDF/A-1b.** If your document fails to pass validation in Step 11 for some reason, then you will have gotten a Preflight report similar to this:


Pages 1 - 127 from "thesis-step10b.pdf"

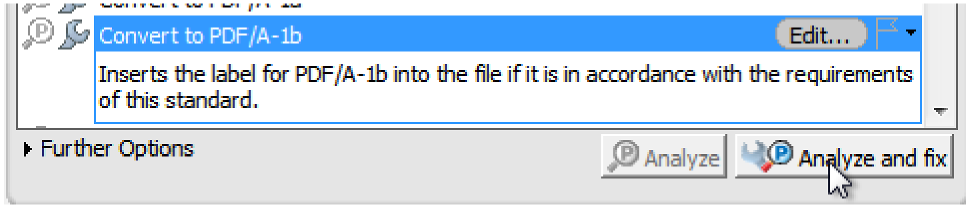
PDF document is not compliant with PDF/A-1b (2005)

Syntax problem: Indirect object "endobj" keyword not preceded by an EOL marker

Syntax problem: Indirect object "obj" keyword not followed by an EOL marker

Syntax problem: Stream dictionary improperly formatted

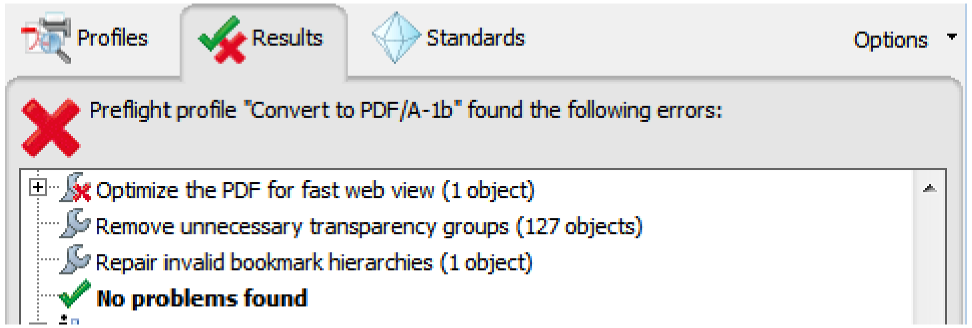
However, there is no cause for concern, as these errors are easily fixed. In Acrobat Pro, go to "View" → "Tools" → "Print Production" → "Preflight" → "Convert to PDF/A-1b" → "Analyze and fix".



Convert to PDF/A-1b

Inserts the label for PDF/A-1b into the file if it is in accordance with the requirements of this standard.

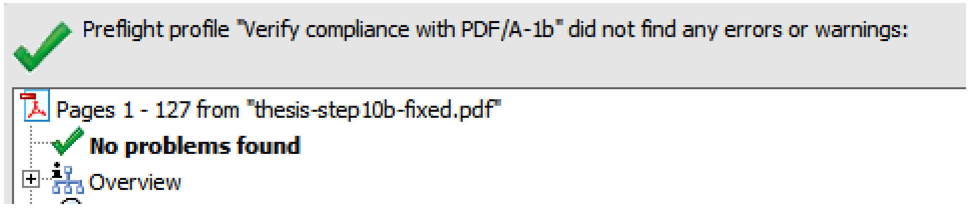
You'll be asked for a filename for the fixed PDF/A file. I chose thesis-step10b-fixed.pdf. If everything goes well, you'll receive a long report showing things that were fixed, and at the end, the message "No problems found".



Preflight profile "Convert to PDF/A-1b" found the following errors:

Optimize the PDF for fast web view (1 object)
Remove unnecessary transparency groups (127 objects)
Repair invalid bookmark hierarchies (1 object)
No problems found

At this point, you can optionally re-run PDF/A-1b validation on the new file, but there will be no further errors.



Preflight profile "Verify compliance with PDF/A-1b" did not find any errors or warnings:

Pages 1 - 127 from "thesis-step10b-fixed.pdf"
No problems found
Overview

Congratulations. If all steps were successful, you should now have a high-quality PDF/A compliant document.

5. Metadata format

5.1. Symbols permitted in metadata

Within the metadata in the `*.xmpdata` file, all printable ASCII characters except `'\'`, `'{'`, `'}'`, and `'%'` represent themselves. Also, all printable Unicode characters from the basic multilingual plane (i.e., up to code point U+FFFF) can be used directly with the UTF-8 encoding. Consecutive whitespace characters are combined into a single space. Whitespace after a macro such as `\copyright`, `\backslash`, or `\sep` is ignored. Blank lines are not permitted. Moreover, the following markup can be used:

```
 \␣              - a literal space  (for example after a macro)
 \%              - a literal '%'
 \{              - a literal '{'
 \}              - a literal '}'
 \backslash      - a literal '\'
 \copyright      - the © copyright symbol
```

The macro `\sep` is only permitted within `\Author`, `\Keywords`, and `\Org`. It is used to separate multiple authors, keywords, etc.

5.2. List of supported metadata fields

Here is a complete list of user-definable metadata fields currently supported, and their meanings. More may be added in the future.

General information:

```
 \Author           - the document's human author. Separate multiple
                     authors with \sep.
 \Title            - the document's title.
 \Keywords         - list of keywords, separated with \sep.
 \Subject          - the abstract.
 \Org              - publishers.
```

Copyright information:

```
 \Copyright        - a copyright statement.
 \CopyrightURL     - location of a web page describing the owner
                     and/or rights statement for this document.
 \Copyrighted      - 'True' if the document is copyrighted, and
                     'False' if it isn't. This is automatically set
                     to 'True' if either \Copyright or \CopyrightURL
                     is specified, but can be overridden. For
                     example, if the copyright statement is "Public
                     Domain", this should be set to 'False'.
```

Publication information:

```
\PublicationType   - The type of publication. If defined, must be
                     one of book, catalog, feed, journal, magazine,
                     manual, newsletter, pamphlet. This is
                     automatically set to "journal" if \Journaltitle
                     is specified, but can be overridden.
\Journaltitle      - The title of the journal in which the document
                     was published.
\Journalnumber     - The ISSN for the publication in which the
                     document was published.
```

```
\Volume            - Journal volume.
\Issue             - Journal issue/number.
\Firstpage         - First page number of the published version of
                     the document.
\Lastpage          - Last page number of the published version of
                     the document.
\Doi               - Digital Object Identifier (DOI) for the
                     document, without the leading "doi:".
\CoverDisplayDate  - Date on the cover of the journal issue, as a
                     human-readable text string.
\CoverDate         - Date on the cover of the journal issue, in a
                     format suitable for storing in a database field
                     with a 'date' data type.
```

See the file sample.xmpdata for an example.

6. Special notes for Dalhousie students

While the above instructions were written for a general audience, here are some additional remarks that may be relevant to students at Dalhousie University.

- Dalhousie has a site license for Acrobat Pro, but it is usually available only to administrators, not students or faculty. Perhaps you can ask a secretary in your department to use his or her copy. If you are a mathematics or statistics student, you can also find a computer running Acrobat Pro XI in room 001 of the Chase building. (Note that of the three computers there, only one has Acrobat Pro XI. Another computer has Acrobat Pro X, which I recommend avoiding if you can).

- The LaTeX template for Dalhousie Theses is available from https://projects.cs.dal.ca/dalthesis/. Please be sure to use version 1.2 or later.

7. Legal notices

7.1. Trademarks

Adobe and Acrobat are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. All other trademarks are the property of their respective owners.

Back to Homepage:

Peter Selinger / Department of Mathematics and Statistics / Dalhousie University
selinger@mathstat.dal.ca / PGP key