Making BachoTEX proceedings — extended version[*]

Jean-Michel HUFFLEN

Abstract

We report an experiment of making the proceedings of a conference by automatically generating as much information as possible. For example, we look for titles and other metadata in successive source files to build the table of contents. Separate source files must be processed by a TEX-like format, which may be pdfLATEX, XƎLATEX, LuaLATEX or ConTEXt. Installing our functions requires a Unix-like `make` command and some programs written mostly using Scheme, partly using Ruby.

0 Introduction

We expose the method we used to make the 3-year proceedings of the Polish-speaking TEX user conference (BachoTEX). These proceedings have some specific aspects, but we think that our method may be applied — even improved — to similar contexts. Our guiding idea: the process of grouping successive articles in one printable file should be automated as far as possible. Besides, we succeeded in avoiding *information redundancy*. For example, the metadata for successive articles — authors, titles and page ranges — are *extracted* from successive source files and the two tables of contents are automatically generated.

In the first section, we introduce to our framework's guidelines and show how our programs behave, we explain why we gained valuable experience through previous projects and why we have volunteered for this task, which should be finished before the summer's end. Section 2 gives the features specific to BachoTEX conferences. Then the successive steps of our *modus operandi* are described in Section 3. Some paths to go further and improve our tools are discussed in Section 4. We give some examples of short programs throughout this article, but reading them requires only basic knowledge in programming.

1 *Genesis*: *Cahiers GUTenberg*, #58

In the fall of 2020, there was some substantial reorganisation of GUTenberg, the French-speaking TEX users group.[1] This process and its reasons have already been described and discussed in many places, including [1, 16], so hereafter we do not give more details about that. We just mention that after a period of lethargy, a new board of directors was elected. During this new board's first meeting, we were appointed as the new editor of this group's journal: the *Cahiers GUTenberg*, as already reported in [13].

When we started our mandate, in the beginning of 2021, we inherited many article proposals, more or less recent, more or less ready. Some used LATEX 2$_\varepsilon$ packages incompatible with other engines, some showed LuaLATEX [11] abilities, some were compilable only by XƎLATEX[2] [28]. In other words, we had to use *several* typesetting systems for composing the same issue. Besides, putting *Cahiers GUTenberg*'s articles on the Web after a while — exactly like *TUGboat*'s — was planned. That is, for an issue, the whole of it will be available, as well as separate articles, including editorials. In addition, let us mention that some `.pdf`[3] files resulting from such articles may be *interactive* or include *animation* effects — e.g., by means of the animate package [10] — when read on line. More precisely, each separate article could be given *printed* and *online* versions — the latter using *hyperlinks* by means of the hyperref package [27] — , additional versions being available if they are of interest. The printed version of the whole issue is easily built by means of the pdfpages package [20].

Controlling the whole process by means of *check programs* may seem perfectionist, or be viewed as a mind game. However, we have personally experienced a situation in which such a control program would have been very useful: the French translation of *The LATEX Companion*'s second edition [22]. As time began to run out, a last change shortened a chapter from two pages: the body's page numbers were updated, but not the page numbers put within the index.[4] We understood that lesson: as soon as we were in charge of the *Cahiers GUTenberg*, we decided to implement programs to automate this process as far as possible — e.g., building the table of contents — other programs are devoted to some checks — the succession of page ranges within the editorial and subsequent articles.

Implementing all our programs by means of additional LATEX 2$_\varepsilon$ classes or additional options of our

[*] A first version of this article [15] has been included in the proceedings of BachoTEX 2024.

[1] In French: ***G****roupe francophone des* ***U****tilisateurs de T**E**X*.

[2] We were told that a source file compiling with XƎLATEX should compile with LuaLATEX. But at this time, that was not always the case; we were forced to acknowledge that as evidence. We asked many people about that; no one was able to explain why to us.

[3] ***P****ortable* ***D****ocument* ***F****ormat*.

[4] This mistake has been pointed out in [26], even if the book's overall review is very good.

doi.org/10.47397/tb/45-2/tb140hufflen-bachoproc

```
(define-record-type <cg-contents-info-record> ; Defining a structure's components for keeping the suitable
  ...)                                        ; information.

(define cg-current-contents-info ; Defining a placeholder for the information to be retained when a .tex file
  '*dummy-value*)                ; is parsed. Processing a new file causes this variable to be reinitialised.

(define cg-toc-function
  (g-mk-tex-parsing-f
   (g-retain-command "author" 1 #f   ; No optional argument.
                     #t              ; Top level only.
                     #f              ; Recursive search if files are input.
                     #t              ; Preamble only.
                     #t              ; Get all the occurrences.
                     (lambda (author-s) ; Function to be applied as soon as this command is to be
                       ...))            ; processed.
   (g-retain-command "title" 2 #t ; The first argument is optional.
                     #t #f #t 1   ; Only one occurrence, then give up.
                     (lambda (ignored-s title-s)
                       (cg-contents-info-set-title! current-contents-info
                                                    (normalize-space title-s))))
   ...))
...
;;; What is launched: processing all the main source files, in turn.
(writeln/return (cg-make-toc cg-source-jobname-list        ; List of main source file pathnames.
                             "issue-58-toc-contents.tex")) ; Pathname for the result.
```

Figure 1: How the table of contents of *Cahiers GUTenberg* #58 has been built.

class for the *Cahiers GUTenberg* would have been possible,[5] but we think that this is *misusing* TEX, because these additional tasks are related to *'actual' programming* more than *typesetting.* As we explained in [13], we think that TEX should not be viewed as a universal multi-task program. TEX's language has the same expressive power as a Turing machine, so theoretically any function can be programmed using TEX's primitives,[6] but as with any specialised language, using it for a purpose other than its intended one may be tedious. The program described in [13], written using the Scheme programming language,[7] aims to do this kind of task: it allows a (LA)TEX source file to be parsed, with particular focus on commands of interest (the rest is just skipped). As an example, extracting the accurate information in order to build the successive items of the table of contents of [2] is sketched in Fig. 1. More details about configuring our (LA)TEX parser in Scheme can be found in [13]. Going back to building a table of contents' successive items, the title is of course given by means of the `\title` command:

```
\title{\foreignlanguage{english}{Donald
  Knuth} : des mathématiques à la
 typographie}
```

The cedram-cg-8 class, designed for *Cahiers GUTenberg*, uses LATEX's predefined command `\author`, but *each* author among several co-authors must be specified by a separate command:

```
\author{\firstname{Sandrine}
 \lastname{Chenevez}}
\author{\firstname{...} \lastname{...}}
...
```

Of course, getting the first and last names of an author is a bit more complicated than getting a title, considered as a whole. We do not detail our method in Fig. 1, but in reality, it is quite long but not very difficult, either. Likewise, getting the first page number is easy. If we are interested in checking the succession of page ranges, the last page number of an article can be got by means of the lastpage package [23], which puts a new label `LastPage`: our program is able to parse an auxiliary (`.aux`) file to catch the information associated with it.

We have just sketched *what* we are doing, but not *when.* Gathering several articles does not mean that they have been finalised w.r.t. the order of appearance; some last minute change may occur, and in some cases, performing all the check procedures

[5] Besides, an option of the previous version of the class used for the *Cahiers GUTenberg* allowed metadata for Web search engines to be generated. The present version does not.

[6] Interested readers can consult [3] about this subject.

[7] A good introductory book to this functional programming language is [30].

Setters:	`\hbsettitleinenglish` `\hbsettitleinpolish`	specify a title's versions	analogous to LaTeX's predefined command `\title`
Getters:	`\hbgettitleinenglish` `\hbgettitleinpolish`	place a version of the title within a text	.... the internal command `\@title`

Table 1: Getters and setters for a title.

again might not be needed. In other words, a program able to determine which procedures are to be run again or have not been run yet is welcome. Historically, the UNIX system's `make` command was the first to automate a sequence of tasks, launching them only if need be [8]. Roughly speaking, a command controlled by `make` is launched if one of its *sources* is more recent than its *target*: in other words, such a command is run if its target does not exist or must be built again. The most frequent case is the production of an executable file from source files. The information concerning sources and targets of such commands is given in a configuration file so-called `Makefile`:

```
target: source1 source2 ...
    command
```

Such a command is useful in software development, especially if a *huge* number of source files are handled. In truth, these configuration files' syntax may be considered old-fashioned and more recent programs, more user-friendly, have appeared, but they are devoted to particular programming languages, most often object-oriented. A '`make`-like' command, specialised for engines and formats built out of TeX is `latexmk` [7]. However, we did not use if because it does not work with ConTeXt; it does not allow a Scheme program to be launched, either.

In addition, let us mention that according to `make`'s terminology, a target may be a generated file's name — e.g., the result of compiling a source file — or just denote a set of actions. 'Traditionally', some particular targets are used to remove useless files, as soon as the installation is performed (`make clean`) or to revert to the state at the installation's beginning (`make distclean`). So such tasks can be integrated into our framework.

When we ended up with these programs, the company in charge of printing and binding this first issue [2] was given:

- the contents (an additional final page was for the company's brand), using the A4 paper size with cropmarks;[8]
- the complete cover, including the spine, rightly dimensioned;

as two `.pdf` files. Generating online or additional versions of separate articles is done by Ruby programs [14], based on *regular expressions*.[9]

2 *Exodus*

In 2018, 2019 and 2023, BachoTeX conferences, organised by the Polish GUST[10] group, did not provide proceedings, unlike what was practised before. Given our experience (cf. §1), we volunteered for this task at BachoTeX 2023's end. Here are the differences, in comparison to *Cahiers GUTenberg* #58:

- we planned a 3-part volume, each part for a conference;
- *two* tables of contents are to be put: a Polish one at the beginning, an English one at the end;
- some authors provide *two* abstracts, in English and Polish: we personally wish to homogenise the layout of such 2-language abstracts;
- authors writing in Polish use different methods for that: the packages babel [4] or polyglossia [6], or an *ad hoc* package polski [24]; as a consequence, the definitions applied to fragments written in Polish should be adaptable;
- here also, typesetting systems depend on articles: an article showing graphical features was based on the PostScript language, other articles use ConTeXt [12].

We have considered that processing ConTeXt source texts is a worthwhile exercise, allowing us to be able to accept such texts for the *Cahiers GUTenberg* in the future. Let us also notice that processing the whole of proceedings by means of ConTeXt is possible, but only if all the articles have been typeset with ConTeXt.

When an article for the *Cahiers GUTenberg* is written in French, the cedram-cg-8 class provides two commands for English and German versions of the title, but only the title in French is included into the table of contents. On the contrary, the articles

[8] *Cahiers GUTenberg*'s first issues used A4. More recent issues' paper size is 165 mm × 225 mm.

[9] Regular expressions can be viewed as an advanced language for describing *patterns* — a good introductory book is [9] — : they are implemented in many programming languages, including Ruby [19]. However, they have not been included into the present standard of Scheme [29].

[10] *Polska **G**rupa **U**żytkowników **S**ystemu TeX*, that is, the Polish-speaking TeX user group.

of BachoTEX conferences may be in English or Polish. The commands we introduce for titles are given in Table 1, they are grouped into *setters* and *getters*, using object-oriented terminology. Such new commands are prefixed by ‘hb’ and have been put within a package so-called hbachotex. The following example illustrates how these commands can be used:

```
\hbsettitleinenglish{%
 Making an Issue of the \cgut}
\hbsettitleinpolish{Skład biuletynu \cgut}
\title{\hbgettitleinenglish\thanks{%
  Title in Polish:
  \foreignlanguage{polish}{%
   \emph{\hbgettitleinpolish}}.}}
```

and it shows that the `\cgut` command should be known when a table of contents is processed, that is, if such a title — in English or Polish — is processed by LATEX again. That has been solved by the following convention within the source `.tex` file:

```
%%%--smlbibtex-plus--[
\newcommand{\cgut}{%
 \foreignlanguage{french}{%
  \emph{Cahiers GUTenberg}}}
%%%--]

\hbsettitleinenglish{...}
```

When a output file is open by our Scheme program, all the fragments surrounded by a starting line:

```
%%%--smlbibtex-plus--[
```

and an ending line:

```
%%%--]
```

can be inserted into this output file.

3 *Leviticus*

We have given an overview of the programs we developed and sketched how they have been written. Now we describe how successive stages are chained. Of course, some errors may cause backtracking within the complete process.

3.1 Creating a master file

First you have to write a *master file*, as sketched in Fig. 2. It is a LATEX source file, but the most important decision, at this step, concerns the articles' succession, as shown in Fig. 2. Such a master file should be located at the root of the directory containing subdirectories for articles; it also loads the hbachotex package, but with a complete option, which gets access to more commands used for finalisation.

When such a file is processed by LATEX, let us notice that some arguments are *useless*: for example, the first arguments of the two commands `\hbinputtoc` and `\hbincludearticleplus`:[11] the former is the language of the table of contents, the latter is the typesetting system to be used.[12] These commands themselves will just process a source file or insert a `.pdf` file, but the information given by its first arguments will be caught by a Scheme program and used to generate tables of contents or separate articles.

```
\documentclass[onecolumn,a4paper]{ltugproc}

...

\setcounter{page}{3}

\begin{document}

\hbinputtoc{polish}{polish-toc}

\hbincludearticleplus{%
 pdflatex}{editorial}{editorial}

\hbpart{Bacho\TeX\ 2018}{}

\hbincludearticleplus{%
 xelatex}{B-2018/hufflen}{hb-greg}

...

\hbpart{Bacho\TeX\ 2023}{}

...

\hbincludearticleplus{%
 lualatex}{B-2023/bolek/foto}{foto}

\hbincludearticleplus{context}{B-2023/egger}{%
 Pocketdiary-modul-2023-tugboatstyle}

...

\hbinputtoc{english}{english-toc}

\end{document}
```

Figure 2: Successive articles for BachoTEX conferences in 2018, 2019, and 2023.

[11] Concerning the `\hbinputtoc` command, its second argument is the file name for the corresponding table of contents. The last two arguments of the `\hbincludearticleplus` command are the directory where source files are located and the job name.

[12] In reality, this `\hbincludearticleplus` command's first argument may be any executable program, provided that it's available in your *path*.

```
polish-toc ...: $(all-source-files)
        $(SCHEME) $(CHECK_MAKE_TOCS) $(all-source-files)

editorial/editorial.pdf: editorial/editorial.tex
        cd editorial ; pdflatex editorial.tex

B-2018/hufflen/hb-greg.pdf: B-2018/hufflen/hb-greg.tex
        cd B-2018/hufflen ; xelatex hb-greg.tex

...
```

Figure 3: Generation of dependencies.

3.2 Playing with a skeleton `Makefile`

1. Our toolbox provides a skeleton file you can use to create your `Makefile`, in which only basic targets — e.g., `clean`, `clean-deps`, `realclean`, `distclean`, ... — and general definitions:

   `SCHEME = ...`

   are specified. Within this configuration file, you can notice an input order for another configuration file, `deps`, not yet existing:

   `+include deps`

   (the `deps` file's contents is to be inserted, but '+' prevents errors if this file does not exist);
2. Run the program `hb-makedepend.scm` — or, better, the `deps` target:

   `make deps`

   — to compute the dependencies from the master file's `\hbincludearticleplus` commands: the result is the `deps` file. Fig. 3 sketches the result corresponding to Fig. 2's example.

3.3 Generating `.pdf` files

1. Now you can generate the separate articles by means of suitable targets, as soon as you make precise starting page numbers.
2. An extended version of Fig. 1's program:

   `hb-maketoc-plus.scm`

   allows all the page ranges of successive parts and articles to be checked. Then the two tables of contents are generated as files input within our master file.
3. The complete file can be built, either by a suitable target of the `Makefile` configuration file:

   `make all`

   or by compiling your master file with pdfLaTeX.

3.4 Post-process

Some targets allow a cover to be built, possibly with a spine, correctly dimensioned. Then you can delete files that have become unnecessary. You could build online versions of separate articles, as we did for the *Cahiers GUTenberg*.

4 *Numbers*

4.1 Requirements

To use our toolbox, you will need, in addition to a working distribution of LaTeX & Co:

- a Scheme compiler (preferred) or interpreter, R7RS-compliant [29]: a good example is Racket [18], but other programs exist;
- a make utility recognising GNU[13] features [31];
- a Ruby interpreter if you are interested in generating variants.

4.2 Discussion

When we work on an article to be included into the *Cahiers GUTenberg* or BachoTeX proceedings, we put or check commands for titles and specify the first page number. Sometimes we also check or adapt multi-language abstracts. Of course, we make sure that an article's source files are compiled correctly;[14] we also allow ourselves to make some slight adjustment if we can avoid errors or warning issued by typesetting systems. At this step, we deal with bibliographies: sometimes a `.bbl` file is provided, sometimes we have to run BibTeX [25] or biber [17].

The dependency relations we generate, as shown in Fig. 3, do not include the production of bibliographies. They do not consider the source files handled by the `\input` command or the graphic files processed by the graphicx package's `\includegraphics` command [5]. Roughly speaking, our dependency relations are sufficient for the operations we perform, but should not be used to make ready articles unfinished.

4.3 Going further

When we adapted our toolbox to the BachoTeX proceedings, the most time-consuming operation was the implementation of a new module for ConTeXt, so-called `s-tugboat-hbachotex.mkxl`, more or less equivalent to the ltugproc LaTeX class. In other contexts, the `\author` command's taxonomy depends

[13] Recursive acronym: GNU's Not UNIX.

[14] We also ask authors to provide the `.pdf` file they got with their installation, if possible.

on classes, as mentioned in §2. However, we would be confident if we had to apply our method in other analogous situations.

In future versions, we plan to study the use of the `latexmk` command. Of course, we will have to propose an alternative for source texts processed with ConTeXt. We could also be able to automate the generation of *metadata* using the `.bib` format for complete proceedings, as well as for separate articles.

5 Conclusion

After some last minor details, we expect *Cahiers GUTenberg* #59 and BachoTeX proceedings to be ready before fall 2024. The future will confirm that or not. Then our programs will be hosted at the servers of the French and Polish groups, as agreed. We think that our system is *open*, because some typesetting systems can coexist, some natural languages, too. We recognise that it is limited because some operations require some experience in programming, especially in Scheme. But we hope that some additional experience will allow us to provide turnkey programs.

Acknowledgements

I thank Karl Berry and Barbara Beeton, who proofread this article. Last but not least, Hans Hagen helped us a lot about aspects related to ConTeXt.

A Additional packages

In this appendix, we briefly mention two packages included in our toolbox.

A.1 The antispam package

This first package allows the definition of commands changing the occurrences of the characters '`@`' and '`.`' within an electronic address. Most often these two characters are substituted respectively by '` (at) `' and '` (dot) `', in which case the `\antispamemail` command of this package may be used. Otherwise, you can put other conventions into action. Notice that if the electronic address is parsed by another command C, we have to define a *robust* command in order for the electronic address to be pre-processed before C is applied. For example, we introduced our electronic address as follows at this article's beginning:

```
    \asnetaddress{jmhuffle@femto-st.fr}
```

the `\asnetaddress` command being defined by:

```
\DeclareRobustCommand{\asnetaddress}[1]{%
 \netaddress{\antispamemail{#1}}}
```

A.2 The more-abstracts package

This second package allows the definition of additional environments for *several* abstracts, in succession and written using different languages. The `abstract` environment is for the English language; other environments are provided for Polish, Spanish, Portuguese. Creating new environments for other languages is possible, too.

References

[1] Jacques André, Patrick Bideault, Denis Bitouzé, Thierry Bouche, Michel Bovani, Maxime Chupin, Daniel Flipo and Yvon Henel: "The Last Decade at GUTenberg". *TUGboat*, vol. 43, no. 1, pp. 7–9. `10.47397/tb/43-1/tb133andre-gut`. 2022.

[2] Association GUTenberg : *Ils sont de retour !*, Vol. 58 de *Cahiers GUTenberg*. `10.60028/cahiers.v2021i58`. Septembre 2021.

[3] Pieter Belmans: *A Turing Machine in LaTeX: the Follow-Up*. 12 December 2010. Universiteit Antwerpen. `pbelmans.ncag.info/blog/2010/12/12/a-turing-machine-in-latex-follow-u`.

[4] Javier Bezos and Johannes L. Braams: *babel. User Guide. Version 24.7*. 26 June 2024. `mirror.ctan.org/macros/latex/required/babel/base/babel.pdf`.

[5] David L. Carlisle and the LaTeX Project: *Packages in the 'Graphics' Bundle*. 22 May 2024. `mirror.ctan.org/macros/latex/required/graphics/grfguide.pdf`.

[6] François Charette, Arthur Reutenauer, Bastien Roucariès and Jürgen Spitzmüller: *polyglossia: Modern Multilingual Typesetting with XeLaTeX and LuaLaTeX*. 15 July 2024. `mirror.ctan.org/macros/unicodetex/latex/polyglossia/polyglossia.pdf`.

[7] John Collins, Evan McLean and David J. Mustliner: *latexmk — Fully Automated LaTeX Document Generation*. 7 April 2024. `ctan.org/pkg/latexmk`.

[8] Stuart I. Feldman: "`Make` — A Program for Maintaining Computer Programs". *Software: Practice and Experience*, vol. 9, no. 1, pp. 255–265. April 1979.

[9] Jeffrey E. F. Friedl: *Mastering Regular Expressions*. 3rd edition. O'Reilly. August 2006.

[10] Alexander Grahn: *The animate Package*. 18 June 2023. `gitlab.com/agrahn/animate`.

[11] Hans HAGEN: "LuaTEX: Howling to the Moon". *Biuletyn Polskiej Grupy Użytkowników Systemu TEX*, vol. 23, pp. 63–68. April 2006.

[12] Tom HOTTEN and Hans HAGEN: *ConTEXt, an Excursion*. Version 990527. 27 May 1999. PRAGMA. `www.pragma-ade.com/general/manuals/ms-cb-en.pdf`.

[13] Jean-Michel HUFFLEN: "Extracting Information from LATEX Source Files". *TUGboat*, vol. 43, no. 2, pp. 136–141. `10.47397/tb/43-2/tb134hufflen-extract`. 2022.

[14] Jean-Michel HUFFLEN: "Making an Issue of *Cahiers GUTenberg*". In: *A Model Kit. Modelling and Implementing Text Typesetting in TEX and Other Systems. Proc. BachoTEX 2023*. April 2023.

[15] Jean-Michel HUFFLEN: "Making BachoTEX Proceedings". In: *Composed Thoughts. Proc. BachoTEX 2023*. May 2023.

[16] Jérémy JUST: "Year 2020 at GUTenberg". *TUGboat*, vol. 42, no. 1, pp. 12–13. `10.47397/tb/42-1/tb130just-gut`. 2021.

[17] Philip KIME and François CHARETTE: *biber. A Backend Bibliography Processor for biblatex. Version biber 2.20 (biblatex 3.20)*. 21 March 2024. `ctan.org/pkg/biber`.

[18] Alexis KING: *An Implementation of R7RS in Racket*. 2015. `github.com/lexi-lambda/racket-r7rs`.

[19] Yukihiro MATSUMOTO: *Ruby in a Nutshell*. O'Reilly. English translation by David L. Reynolds, Jr. November 2001.

[20] Andreas MATTHIAS: *The pdfpages Package*. 29 May 2024. `ctan.org/pkg/pdfpages`.

[21] Frank MITTELBACH and Michel GOOSSENS, with Johannes BRAAMS, David CARLISLE, Chris A. ROWLEY, Christine DETIG and Joachim SCHROD: *The LATEX Companion*. 2nd edition. Addison-Wesley Publishing Company, Reading, Massachusetts. August 2004.

[22] Frank MITTELBACH et Michel GOOSSENS, avec Joannes BRAAMS, David CARLISLE, Chris A. ROWLEY, Christine DETIG et Joachim SCHROD : *LATEX Companion, 2^e^ édition*. Pearson Education France. Traduction française de [21] par Jacques ANDRÉ, Benoît BELLET, Jean-Côme CHARPENTIER, Jean-Michel HUFFLEN et Yves SOULET. Octobre 2005.

[23] H.-Martin MÜNCH and Jeffrey P. GOLDBERG: *The lastpage Package*. 3 July 2024. `ctan.org/pkg/lastpage`.

[24] Mariusz OLKO and Marcin WOLIŃSKI: *Pakiet polski, wersja 1.3.6*. 25 August 2021. `ctan.org/pkg/polski`.

[25] Oren PATASHNIK: *BIBTEXing*. February 1988. Part of the BIBTEX distribution. `ctan.org/pkg/bibtex`.

[26] Éric PICHERAL : « Compte-rendu de lecture ». *La Lettre GUTenberg*, Vol. 31, p. 13–14. Janvier 2006. `www.gutenberg-asso.fr/Lettre-GUTenberg-31`.

[27] Sebastian RAHTZ, Heiko OBERDIEK and THE LATEX PROJECT: *Hypertext Marks in LATEX: a Manual for hyperref*. 10 July 2024. `mirror.ctan.org/macros/latex/contrib/hyperref/doc/hyperref-doc.pdf`.

[28] Will ROBERTSON, Khaled HOSNY and Karl BERRY: *The XƎTEX Reference Guide*. 1 March 2024. `ctan.org/pkg/xetexref`.

[29] Alex SHINN, John COWAN, and Arthur A. GLECKLER, with Steven GANZ, Aaron W. HSU, Bradley LUCIER, Emmanuel MEDERNACH, Alexey RADUL, Jeffrey T. READ, David RUSH, Benjamin L. RUSSEL, Olin SHIVERS, Alaric SNELL-PYM and Gerald Jay SUSSMAN: *Revised^7 Report on the Algorithmic Language Scheme*. 6 July 2013. `trac.sacrideo.us/wg/raw-attachment/wiki/WikiStart/r7rs.pdf`.

[30] George SPRINGER and Daniel P. FRIEDMAN: *Scheme and the Art of Programming*. The MIT Press, McGraw-Hill Book Company. 1989.

[31] Richard M. STALLMAN, Roland MCGRATH and Paul D. SMITH: *GNU Make. A Program for Directing Recompilations. Version 4.4.1*. February 2023. `www.gnu.org/software/make/manual/make.pdf`.

◇ Jean-Michel HUFFLEN
FEMTO-ST (UMR CNRS 6174)
& University of Bourgogne
Franche-Comté
16, route de Gray
25030 BESANÇON CEDEX
FRANCE
`jmhuffle (at) femto-st (dot) fr`
`https://members.femto-st.fr/Hufflen-Jean-Michel/en`