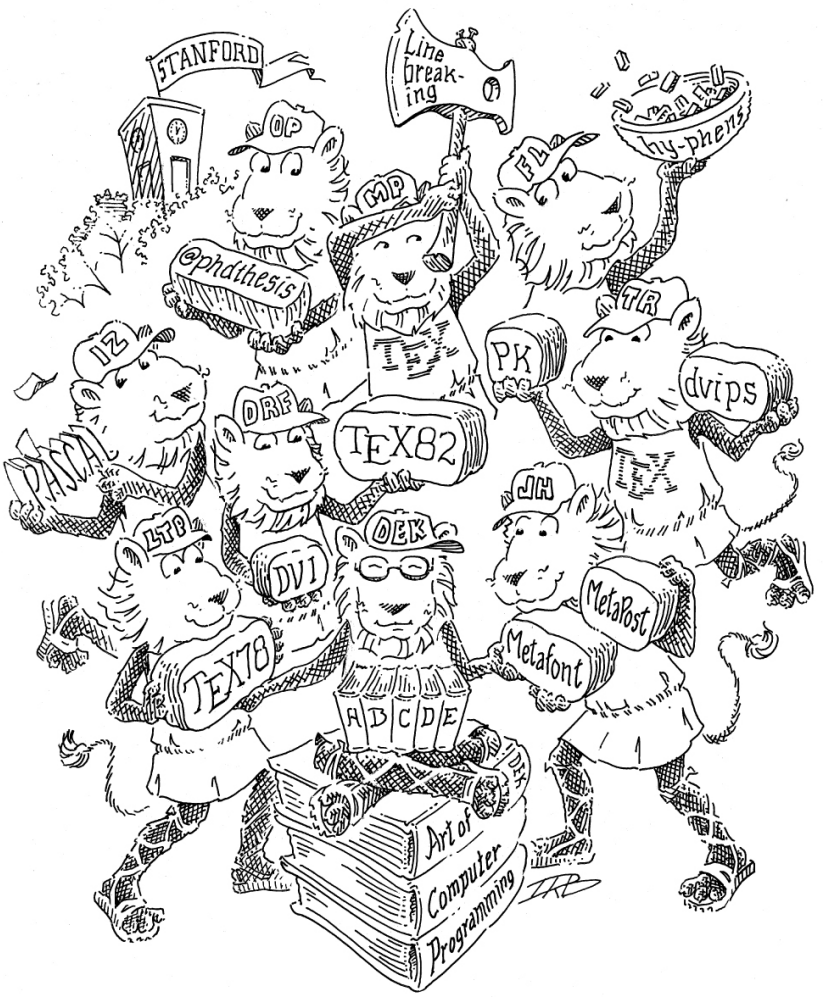
TEX's 2^5 Anniversary

A Commemorative Collection

TEX's 2^5 Anniversary: A Commemorative Collection

Editors
KARL BERRY
DAVID WALDEN

Illustrations
DUANE BIBBY



TEX Users Group
Portland, Oregon

TEX Users Group, Publishing
P.O. Box 2311
Portland, OR 97208-2311
USA

ISBN 978-0-9824626-1-4
First printing, 2010

To Donald Knuth
and his Stanford collaborators
in creating the world of TEX

Contents

Preface

Donald Knuth's dissatisfaction with the typesetting of the second edition of one of the volumes of his magnum opus *The Art of Computer Programming* and his consequent decision to develop a new computer-based typesetting system is well known. As is his practice, Professor Knuth comprehensively documented the need for better typesetting ("Mathematical Typography," reprinted as chapter 2 of *Digital Typography*, Donald Knuth, CLSI Publications, Stanford University, 1999). He has also retroactively published some of his earliest notes on the system he began developing in 1977 and first released in 1978 (*Digital Typography*, chapters 24 and 25).

As a Stanford faculty member, Prof. Knuth had students who could join in his research and share in the design and development work. Chapter 25 of *Digital Typography* mentioned above, written in 1977, already mentions turning some implementation work over to Michael Plass and Frank Liang. These students were the first in a succession of contributors at Stanford who were close to Prof. Knuth and helped in one way or another to implement key parts of TEX.

With this annual TUG conference in San Francisco celebrating the 2^5 (thirty-second) anniversary of TEX, Prof. Knuth and other Stanford TEX developers were invited to participate in the conference, and kindly accepted. We tried to think of a suitable gesture with which to honor them, and the creation of this commemorative volume of reprints from *TUGboat* seemed most appropriate.

To set the context, we invited one other person who has been in the TEX world since nearly the beginning to contribute a foreword: Barbara Beeton, from the American Mathematical Society.

As listed in the preceding table of contents, we have included three papers by Prof. Knuth, one Q&A session with him, and one paper each from eight of his collaborators. Knuth's papers are Part 1 of this volume. The other contributors' papers make up Part 2, which is ordered alphabetically by the authors' last names.

The approximate chronological order of the papers is as follows, with our chapter number listed at the beginning of the line:

11.	Zabala & Trabb Pardo on the PASCAL implementation status	October 1980
5.	Fuchs on DVI format	July 1981
7.	Liang on TEX and hyphenation	July 1981
9.	Plass on applying TEX to grammar charts	November 1981
1.	Knuth on (then) current and future plans for TEX	October 1981
12.	Trickey & Curtis on porting TEX to VAX/UNIX	April 1983
10.	Rokicki on PK packed file format	November 1985
8.	Patashnik on BIBTEX	June 1998
6.	Hobby on MetaPost	December 1989
2.	Knuth on completing his TEX/METAFONT work	November 1990
3.	Knuth question & answer session in Oxford	September 1999
4.	Knuth on his most recent TEX/METAFONT tuneup	May 2008

We mostly preserved formatting details, instead of trying to unify all these disparate sources produced over a three-decade time span. The main difference from the originals is the line length of this book (compared to *TUGboat*), and general usage of the current Computer Modern fonts. We have not given the authors any opportunity to correct anything they might now like to change or correct in the papers.

Finally, we have scattered images created over the years by Duane Bibby throughout the book. Duane also created a new image for the front cover. The very last drawing in the book, accompanying the article by Howard Trickey, was also created for the book — a late-breaking bonus to try to make up for our unwitting omission of Howard from the cover drawing.

Thanks to Barbara Beeton not only for her gracious foreword, but also for her superlative editing skills (as always); to Hans Hagen for supplying more Bibby drawings at the last minute; to the Computer Science Department at the University of Aarhus in Denmark, which supports the main TUG server, where we developed this book; and to Robin Laakso, TUG's executive director, who organized the TUG 2010 conference and supported us in many ways during the creation of this book.

Karl Berry & David Walden
April 2010

There is some confusion about what is
a preface, a foreword, and an introduction.
The terms are often interchanged
without any apparent reason.

— Marshall Lee, *Bookmaking: The Illustrated Guide to Design and Production* (1965)

Part I
Donald Knuth

It is a cliche to say "He needs no introduction", but it is nevertheless true when we are speaking of Donald Knuth in a book addressed to the TEX community.

We have chosen four items from Professor Knuth's many publications in *TUGboat*: (1) his 1981 announcement of the impending "definitive" release of TEX, four years after he began system design in 1977; (2) his 1990 announcement that he was done working on TEX except for periodically fixing "extremely serious bugs"; (3) a Q&A session held in Oxford in 1999; (4) his report on his 2008 tuneup of TEX.

In some sense, these papers summarize Professor Knuth's attitude toward TEX over the past 32 years. They also sketch the history of what to our knowledge is the longest running free software/open source development story and Professor Knuth's approach to making that possible. The Q&A illustrates Knuth's commitment to not only doing extraordinary work but explaining it, in his own inimitable style. Finally, the last paper shows that even when merely doing a tuneup Professor Knuth never departs from his enthusiastic and precise approach to the art of computer programming and analysis of algorithms.

Bibliographic information on Professor Knuth is also available from other sources: a list of interviews of him at `http://tug.org/interviews`, his introductions in his seven collections of papers in different areas of his research, and fan web sites dedicated to him.

TUGboat bibliography

The current state of things [**2**:3, November 1981]

Fixed-point glue setting — an example of WEB [**3**:1, March 1982]

A note on hyphenation [**4**:2, September 1983]

Observations on TEX from a divergent viewpoint: Comments, response, and reresponse [**4**:2, September 1983]

TEX incunabula [**5**:1, May 1984]

Comments on quality in publishing [**5**:1, May 1984]

A course on METAFONT programming [**5**:2, November 1984]

Recipes and fractions [**6**:1, March 1985]

Remarks to celebrate the publication of *Computers & Typesetting* [**7**:2, June 1986]

The TEX logo in various fonts [**7**:2, June 1986]

It happened: Announcement of TEX 2.1 [**8**:1, April 1987]

Mixing right-to-left texts with left-to-right texts (with Pierre MacKay) [**8**:1, April 1987]

Problem for a Saturday morning [**8**:1, April 1987]

Fonts for digital halftones [**8**:2, July 1987]

Problem for a Saturday Morning — A Solution [**8**:2, July 1987]

Reply: Printing Out Selected Pages [**8**:2, July 1987]

Macros for Jill [**8**:3, November 1987]

... the designer of a new system
must not only be the implementor and first large-scale user:
the designer should also write the first user manual.

— DONALD KNUTH, *Software — Practice and Experience* (1989)

Part II
Early participants

In this part of the book we reproduce one paper by each of (alphabetically) David Fuchs, John Hobby, Frank Liang, Oren Patashnik, Michael Plass, Tomas Rokicki, Luis Trabb Pardo, and Howard Trickey, most of whom we expect to be present with Donald Knuth at the 2010 TUG annual conference in San Francisco. Each has a chapter with a brief biographical note, the bibliography of their papers from *TUGboat*, and the reprinted paper.

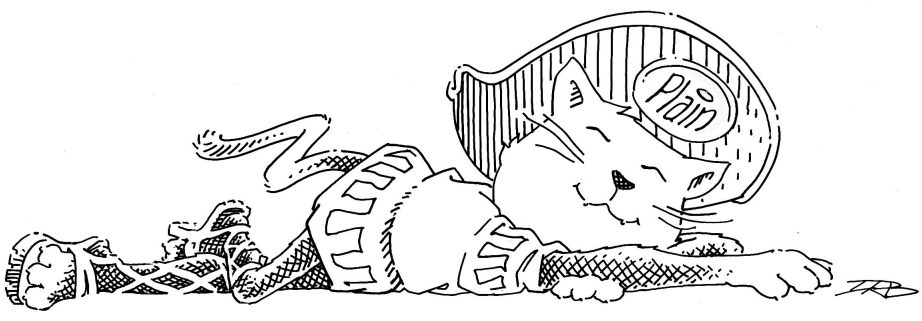
Ignacio Zabala is the first author of the Trabb Pardo paper (page 85). We were unable to contact him and, thus, regretfully could not invite him to the 2010 conference. Zabala has no other papers in *TUGboat*; his Ph.D. dissertation was *Interacting with Graphic Objects*, and his advisor was Donald Knuth. Zabala made a major contribution in those early days of TEX as described in the paper with Trabb Pardo and in the following quotes by Knuth.

There were also various calligraphers and font designers involved at Stanford during those early days of TEX.

Hermann Zapf's influence on Knuth and his involvement with Knuth has been well documented in several papers, for example, those reproduced in Knuth's 1999 *Digital Typography* book.

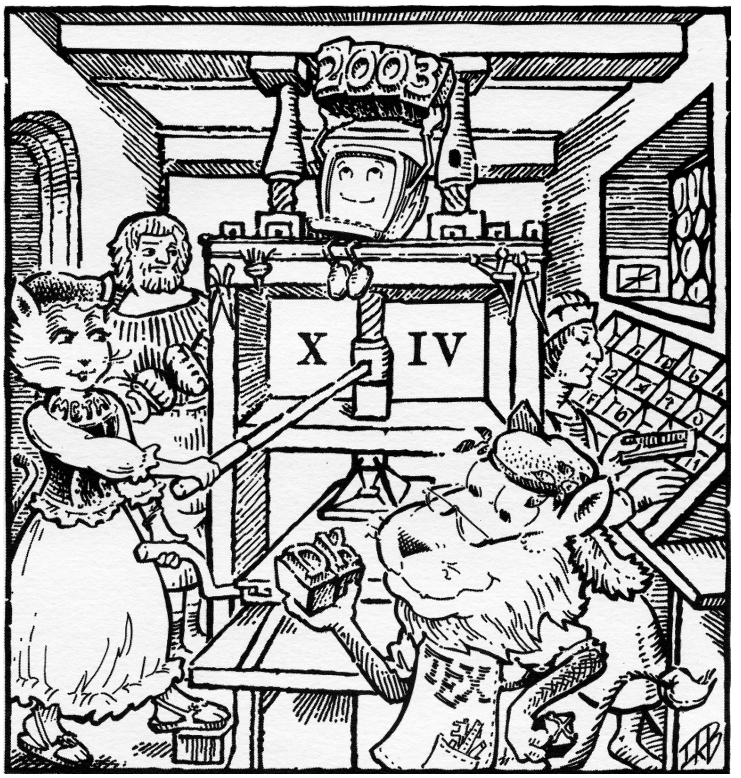
Knuth has often acknowledged the help and influence of font designer Matthew Carter, though he was at Stanford only for a short time. Knuth has also noted Gerard Unger, who spent February 1985 at Stanford.

Charles Bigelow, Kris Holmes, and Richard Southall also worked close by and influenced Knuth or were involved with his font- and METAFONT development efforts. Bigelow started a Master's program in digital typography at Stanford. He was on Stanford's faculty for thirteen years and taught type design, typography, and the history and theory of writing. In a 1984 *TUGboat* article, "A course on METAFONT programming," Knuth describes a course that he jointly taught with Bigelow and Southall which was attended by what he calls "four dozen brave students."



7

Frank Liang



By 1977 Franklin Liang was already involved with Donald Knuth and TEX. According to the Acknowledgements in his thesis (`http://tug.org/docs/liang`), Frank began his involvement with TEX as a summer job. In Frank's paper reproduced in this chapter, he states that the hyphenation algorithm described in the paper "was developed by Prof. Knuth and myself in the summer of 1977." We also know that in the summer of 1977 Donald Knuth revised `TEXDR.AFT` [*Digital Typography*, pp. 481–504] to be `TEX.ONE` [*Digital Typography*, pp. 505–532] so his research assistants Frank Liang and Michael Plass could prepare a prototype implementation while he was away on a trip.

The original implementation of hyphenation in TEX was changed for TEX82, using the pattern-based method hinted at in the final short section of the paper reproduced here, which was also the subject of Frank's thesis dated August 1983. This method is also described in Appendix H of *The TEXbook*.

We also know that during his time at Stanford, Frank did some other mathematical work: "A Lower Bound for On-Line Bin Packing," *Information Processing Letters 10*(2), pp. 76–79, 1980; *The Dinner Table Problem* (with Bengt Aspvall, `ftp://db.stanford.edu/pub/cstr.old/reports/cs/tr/80/829/CS-TR-80-829.pdf`). In any case, Frank's pattern-based method for hyphenation is essentially statistical. In 1979 Frank was a teaching assistant for Knuth's Concrete Mathematics course which was given that year by Ron Graham.

Frank's pattern-based approach to hyphenation essentially solved the hyphenation problem for TEX and has been used in many other free software document processors, such as OpenOffice and Apache FOP. (The method may also be used in various commercial systems.) A 2007 document on *troff* (`http://heirloom.sourceforge.net/doctools/troff.pdf`) credits Frank's approach to hyphenation.

By now, sets of patterns have been produced for essentially all languages that use hyphenation (see, for instance, `http://ctan.org/tex-archive/language/hyph-utf8`).

TUGboat bibliography: The paper included here is Frank's only item in *TUGboat*.

TEX and hyphenation

[Published in *TUGboat* **2**:2, July 1981]

Word hyphenation is a useful feature of any computerized document formatting system. Sometimes it is also one of the most embarrassing.*

The current hyphenation algorithm was developed by Prof. Knuth and myself in the summer of 1977. Our goal was to come up with a reasonably compact algorithm that would find a significant percentage of possible hyphenation points, but would make very few errors. The algorithm is described in Appendix H of the TEX manual. Note that there have been quite a few minor changes since the original printing of the manual; these are described in the errata file.

Basically, the algorithm has four types of rules: (1) Prefix removal (e.g. `com-`, `dis-`, `ex-`), (2) Suffix removal (e.g. `-able`, `-ful`, `-tion`), (3) Vowel-consonant-consonant-vowel rule (can usually split between the consonants), and (4) Exception table (about 300 entries). Actually, these parts are applied in the order (4), (2), (1), (3); this order is rather important because of the interaction between rules. For example, the `horse-` prefix was put in not so much because we were concerned about hyphenating words like `horse-power` correctly, but rather to avoid hyphenating them incorrectly (the vccv rule (3) would break `hor-sepower`).

The rules were mostly found by hand. Good prefixes were found by looking through a dictionary; suffixes by looking through a reverse dictionary. Other ad hoc rules were discovered as the development proceeded (break vowel-q, break after ck). However, as good computer scientists, we then used an on-line copy of the American Heritage Dictionary (at Xerox PARC) to test our rules. This testing had two purposes: (1) to determine which pairs of consonants should be split under the vccv rule, and (2) to generate a list of exceptions to the rules. The exception list originally contained thousands of words, but was pruned down to just a few hundred. Also, in some cases new rules were formulated to take care of large classes of exceptions.

How well does the algorithm work in practice? Quite well, it seems. Quantitatively, in a test on a pocket dictionary word list, the algorithm found about 40% of the allowable hyphen points, with about 1% in error. Furthermore, the hyphen points found were usually the most reasonable or "good" places to break the word. In practice, the algorithm almost never makes a glaring mistake, while at the same time the user does not very often need to specify explicit (discretionary) hyphens, unless the columns are very narrow (or letters very wide).

The algorithm takes about 4K 36-bit words of code, including the exception dictionary.

A note on the implementation: If the algorithm is programmed by sequentially checking each of the rules to see if it applies, it will run rather slowly. Using a hash table would improve things,, but a faster and more compact way is to use a version of a finite state machine. Interested readers should look at the actual code.

Time magazine algorithm

This is reputedly the most widely used hyphenation algorithm (of acceptable quality). The idea is to decide whether or not to split a word based on four letters `wx-yz`

*If you find any such embarrassing hyphenations done by TEX, you are encouraged to send them to the author.

ISBN 978-0-9824626-1-4
90000
9 780982 462614

TEX USERS GROUP