

Neue BIB_TE_X-Style-Files: Die *adaptable family*

Hans-Hermann Bode*

08. April 1992

Zusammenfassung

Es wird eine neue Familie von BIB_TE_X-Style-Files vorgestellt, die von den Standard-Style-Files abgeleitet ist und sich durch nachträgliche Anpaßbarkeit von Layout und Füllwörtern seitens des Benutzers oder der Benutzerin¹ auszeichnet. Das Paket umfaßt neben den eigentlichen Stilen Dateien mit Definitionen, die automatisch eine englische oder deutsche Bibliographie mit ausgeschriebenen oder abgekürzten Füllwörtern erzeugen und außerdem die volle Kompatibilität bestehender BIB_TE_X-Datenbanken zu den neuen Stilen sowie umgekehrt die Verträglichkeit neuer Datenbanken zu den alten Stilen herstellen.

Mit BIB_TE_X steht dem L^AT_EX- wie dem plain-T_EX-Anwender ein außerordentlich leistungsfähiges Werkzeug zur Verfügung, um Bibliographien zu Artikeln, Berichten, Büchern und *literate programs* zu erstellen. Mußte man früher mühsam alle zitierten Literaturstellen zusammensuchen, alle Angaben – oft immer wieder dieselben – per Hand eintippen (möglichst, ohne Fehler bei Seitenangaben, Bandnummern, Jahreszahlen usw. zu machen) und sich schließlich auch noch um Layout-Fragen kümmern, so beschränkt sich diese Arbeit mit BIB_TE_X auf die einmalige Eingabe der bibliographischen Informationen in eine Literaturdatenbank und das Aufrufen von `\cite`-Kommandos mit einem Schlüssel für das zu zitierende Werk. BIB_TE_X stellt dann die Literaturliste zusammen und sorgt auch für ein konsistentes Aussehen derselben.

BIB_TE_X ist „offizieller“ Bestandteil des L^AT_EX-Makropakets und von L^Amport in [3] beschrieben, eine deutsche Fassung findet man etwa bei Kopka [2]. Darüberhinaus erlaubt das zur Standard-BIB_TE_X-Distribution gehörige Makropaket `btmac.tex` auch die Verwendung unter plain-T_EX und zwar mit der gleichen Benutzerschnittstelle (dieses Makropaket ist auch mit dem WEB-System für strukturierte Dokumentation kompatibel, so daß BIB_TE_X auch beim *literate programming* eingesetzt werden kann). Oren Patashnik, der Autor des BIB_TE_X-Programms, legt mit [4] ein Manual für BIB_TE_X-Benutzer vor, das im Gegensatz zu den zuvor genannten Stellen die jeweils aktuelle Version dokumentiert und das in einer Datei `btmdoc.tex` jeder BIB_TE_X-Implementation beigegeben sein sollte. Die Kenntnis wenigstens einer der angegebenen Quellen wird hier vorausgesetzt und nicht weiter auf allgemeine Grundlagen der Benutzung von BIB_TE_X eingegangen.

Das Schlußkapitel von Patashniks Manual – herausgegeben als separates Dokument [5] – beschreibt Möglichkeiten der Erzeugung von Style-Files, die das Layout der Bibliographie bestimmen. Vier solcher Style-Files (`plain`, `unsrt`, `alpha` und `abbrv`) gehören zum Standardumfang von BIB_TE_X. Weitere Styles – wie `siam`, `ieeetr` oder `apalike` – wurden von Patashnik u. a. erstellt, in erster Linie, um den Anforderungen bestimmter Zeitschriftenherausgeber zu entsprechen. Die Erzeugung von solchen Styles, auch wenn es nur um relativ geringfügige Abänderungen vorhandener Stile geht, ist allerdings ein recht mühsamer Vorgang (man muß Programme in einer Spezialsprache mit Postfixnotation schreiben), der die meisten BIB_TE_X-Anwender überfordern dürfte. Aus diesem Grund entstand die *adaptable family*, die eine einfache Anpassung an persönliche oder fremde Vorstellungen erlaubt.

Die Styles

Das grundsätzliche Aussehen der Bibliographie wird auch bei den anpaßbaren Versionen durch den jeweiligen, im `\bibliographystyle` der T_EX-Quelle gewählten Stil festgelegt. Dieser bestimmt im wesentlichen, welche Informationen überhaupt auszugeben sind und in welcher Reihenfolge diese angeordnet werden sollen; darüber

*Postanschrift: Arbeitsgruppe Systemforschung, Universität Osnabrück, Artilleriestraße 34, D-W-4500 Osnabrück; e-mail: `hhbode@dosuni1.bitnet`.

¹Im folgenden gilt die Fußnote auf Seite v von Knuths fundamentalem Werk [1] sinngemäß.

Tabelle 1: Übersicht der Stile mit ihren Attributen.

Stil	numerisch	sortiert	initial	klein	kurz	titellos
aplain	ja	ja	nein	ja	nein	nein
aunsrt	ja	nein	nein	ja	nein	nein
aalpha	nein	ja	nein	ja	nein	nein
aabbrv	ja	ja	ja	ja	ja	nein
anotit	ja	ja	ja	ja	ja	ja
aunsnot	ja	nein	ja	ja	ja	ja

Tabelle 2: Zusammenstellung der verwendeten Makros.

Makroname	Bedeutung
\abtype	Definiert die Schriftauszeichnung einzelner Felder eines Bibliographieeintrags.
\abphrase	Legt die Füllwörter fest.
\abmonth	Enthält die Monatsnamen.
\abedition	Stellt Bezeichnungen für Auflagen eines Buches zur Verfügung („Erste“, „Zweite“, „Dritte“ usw.).
\abchapter	Gibt Namen für bestimmte Segmente eines Schriftstücks zurück („Abschnitt“, „Absatz“, „Anhang“ oder „Teil“).

hinaus sind für jeden Stil eine Reihe von festen, unveränderlichen Merkmalen vorgegeben, die in Tabelle 1 für die sechs Stile der *adaptable family* zusammengefaßt sind.

Alle Stile bis auf **aalpha** erzeugen numerische Marken zur Kennzeichnung der Zitate („[1]“, „[2]“ usw.), während **aalpha** die Marken alphanumerisch, i. allg. aus Kürzeln für die Autorennamen und Erscheinungsjahr, zusammensetzt (z. B. „[knu73]“). Bei **aunsrt** und **aunsnot** erscheint die Bibliographie in der Reihenfolge der Zitate, bei den anderen Stilen wird sie nach Autorennamen, Erscheinungsjahren usw. sortiert. Das Attribut „initial“ zeigt an, ob die Vornamen der Autoren abgekürzt werden oder nicht; das erstere ist der Fall für **aabbrv**, **anotit** und **aunsnot**, die zudem im Unterschied zu den übrigen Stilen als „kurz“ gekennzeichnet sind (die Bedeutung dieses Attributs wird im nächsten Abschnitt erklärt). Alle Stile setzen Großbuchstaben, die in Titeln von Zeitschriftenartikeln – sofern diese überhaupt ausgegeben werden – vorkommen und nicht am Satzanfang stehen, in Kleinschreibung um (Attribut „klein“); dies ist zwar für deutsche Titel nicht geeignet, läßt sich aber leicht unterdrücken (s. die Erörterung am Ende dieses Artikels). Schließlich zeichnen sich die Stile **anotit** und **aunsnot** dadurch aus, daß die Titel von Zeitschriftenartikeln unterdrückt werden, wie es etwa in der physikalischen Literatur durchgängig üblich ist.

Insgesamt entsprechen die Stile **aplain**, **aunsrt**, **aalpha** und **aabbrv** – wie auch aus der Namensgebung hervorgeht – den vier Standardstilen von BIB_{TEX} (Patashniks *plain family*), während **anotit** und **aunsnot** hinzugefügt wurden, um eine Lücke zu schließen.

Anpassung von Layout und Füllwörtern

Die Anpassung der **a**-Stile erfolgt durch Definition bestimmter TEX -Makros; andererseits funktionieren die Stile nicht, wenn diese Makros undefiniert sind. Die Definition geschieht zweckmäßigerweise über ein $\text{C}\text{P}\text{R}\text{E}\text{A}\text{M}\text{B}\text{L}\text{E}$ -Kommando, das sich in einer der bibliographischen Datenbanken befindet (vgl. das nächste Kapitel). Die **a**-Styles definieren einen Schalter **abfull**, der in Abhängigkeit davon, ob es sich um einen „Kurzstil“ handelt oder nicht (s. Tab. 1), den Wert **false** oder **true** bekommt. So können sich die Makros automatisch darauf einstellen, Füllwörter abzukürzen oder voll auszuschreiben.

Die fraglichen Makros sind in Tab. 2 zusammengestellt. Alle Makronamen beginnen mit **ab**. Während **\abtype** für die Schriftauszeichnung verantwortlich ist, haben alle anderen Makros mit Füllwörtern zu tun und erlauben sprachspezifische Anpassungen sowie Unterscheidung zwischen „Kurz-“ und „Vollstilen“. Es folgen detaillierte Beschreibungen der Makros mit ihren Parametern.

Tabelle 3: Der erste Parameter von `abtype`.

Code	Bedeutung
0	Namen von Autoren und Editoren
1	Titel von Büchern und Reihen
2	Zeitschriftentitel
3	Bandnummern von Zeitschriften
4	Heftnummern von Zeitschriften
5	Datumsangaben

abtype

Die Layoutsteuerung wird über das Makro `\abtype` vorgenommen. Die a-Stile schreiben die meisten Felder eines Eintrags indirekt über einen Aufruf von `\abtype` in die Bibliographie, wobei zwei Argumente übergeben werden: eine Codennummer für die Art des Feldes und der in das Feld einzutragende Text. Dieser kann dann vom Makro beliebig formatiert zurückgegeben werden. Die Codewerte für den ersten Parameter von `\abtype` findet man in Tab. 3.

Durch eine Definition der Form

```
\def\abtype##1##2{%
\ifcase##1{\sc##2}\or"##2"\or{\em##2/}\or{\bf##2}\or%
(##2)\or(##2)\else##2\fi}}
```

also erreichte man beispielweise, daß Autorennamen in KAPITÄLCHEN, Buchtitel in „Anführungszeichen“ und Zeitschriftentitel *kursiv* gesetzt werden und daß Bandnummern von Zeitschriften **fett** erscheinen, während Heftnummern ebenso wie Datumsangaben geklammert werden. Vorsichtshalber wurde noch eine `\else`-Klausel angefügt, die bewirkt, daß bei Aufruf des Makros mit einem undefinierten Codewert das zweite Argument unformatiert zurückgegeben wird (für die Original-a-Stile wäre dies nicht erforderlich).

abphrase

Der größte Teil der Füllwörter wird durch `\abphrase` abgedeckt. Das Makro hat einen Parameter, der wieder eine Codennummer darstellt. Die vorkommenden Werte zeigt Tab. 4.

Ein Beispiel für eine geeignete Definition von `\abphrase` mit ausgeschriebenen, deutschen Füllwörtern wäre:

```
\def\abphrase##1{%
\ifcase##1{ und }\or{ }\or{ und andere}\or%
{ (Herausgeber)}\or{ (Herausgeber)}\or%
{ aus }\or{ in }\or{In }\or%
{Band}\or{Band}\or{Nummer}\or{Nummer}\or%
{ Auf" |lage}\or{Seiten}\or{Seite}\or{Kapitel}\or%
{Bericht}\or{Diplomarbeit}\or{Dissertation}\fi}}
```

abmonth

Einfacher als die vorangegangenen Makros ist `\abmonth`: Es wird mit einer Zahl zwischen 1 und 12 als Argument aufgerufen und sollte die Bezeichnung für den entsprechenden Monat zurückgeben; also etwa

```
\def\abmonth##1{\ifcase##1\or Januar\or Februar\or M"arz\or
April\or Mai\or Juni\or Juli\or August\or September\or
Oktober\or November\or Dezember\fi}}
```

Man beachte, daß vor `Januar` bereits ein `\or` stehen muß, da dieser Monat den Code 1 hat und nicht 0. Desweiteren gilt, daß die Definition von `\abmonth` nur dann einen Effekt hat, wenn in den Bibliographiedatenbanken von den vordefinierten Strings `jan`, `feb`, ..., `dec` Gebrauch gemacht wird – es sei denn, man ruft das Makro direkt auf, wovon man jedoch im Hinblick auf mögliche Kompatibilitätsprobleme mit anderen Styles oder zukünftigen Versionen der anpaßbaren Stile Abstand nehmen sollte.

Tabelle 4: Der Parameter von `abphrase`.

Code	Bedeutung
0	Bindewort zwischen zwei oder vor dem letzten Namen einer Liste („und“)
1	Satzzeichen vor dem letzten Bindewort einer Namensliste (im Deutschen nichts, im Englischen ein Komma)
2	Ersatz für Namen von Mitautoren und Mitherausgebern („u. a.“ oder „et al.“)
3	Bezeichnung für mehrere Herausgeber (z. B. „Editoren“)
4	Bezeichnung für einen Herausgeber („Editor“)
5	Wort nach der Bandnummer einer Reihe oder Kreuzreferenz („aus“)
6	Wort nach der Ausgabennummer einer Reihe („in“)
7	Wort vor Querverweisen auf Bücher, Artikel usw. („In“)
8	Wort vor einer Bandnummer mitten im Satz („volume“ auf Englisch, aber „Band“ auf Deutsch)
9	dto. am Satzanfang („Volume“ bzw. erneut „Band“)
10	Wort vor einer Reihenummer mitten im Satz („Nummer“)
11	dto. am Satzanfang (im Deutschen dasselbe)
12	Wort hinter einer Auflagennummer („Auflage“)
13	Wort vor mehreren Seitennummern („Seiten“)
14	Wort vor einer einzelnen Seitennummer („Seite“)
15	Wort vor einer Kapitelnummer („Kapitel“)
16	Bezeichnung für einen Forschungsbericht („Technical Report“)
17	Bezeichnung für eine Diplomarbeit („Master’s thesis“)
18	Bezeichnung für eine Doktorarbeit („PhD Thesis“)

abedition

Ein weiteres Makro mit einem numerischen Wert als Parameter ist `\abedition`. Die übergebene Zahl bezeichnet eine Auflagennummer und es sollte ein entsprechender String zurückgegeben werden, z. B.:

```
\def\abedition##1{\ifcase##1\or Erste\or Zweite\or Dritte\or
  Vierte\or F"unfte\or Sechste\or Siebte\or Achte\or
  Neunte\or Zehnte\else?\fi}
```

In den anpaßbaren Stilen existieren vordefinierte Strings `first`, `second`, ..., `tenth`, die man aus den gleichen Gründen wie oben für das `EDITION`-Feld der Bibliographieeinträge verwenden sollte. Ein Problem ergibt sich, wenn eine Auflagennummer größer als 10 benötigt wird; in diesem Fall sollte man das Makro entsprechend erweitern und zusätzliche Strings durch `@STRING`-Kommandos definieren.

abchapter

Schließlich ist noch ein Makro mit einem Code zwischen 0 und 3 für Bezeichnungen verschiedener Segmente eines Dokuments zu definieren. Die Bedeutung ergibt sich unmittelbar aus dem Beispiel

```
\def\abchapter##1{\ifcase##1Abschnitt\or Absatz\or
  Anhang\or Teil\fi}
```

Auch für den Aufruf dieses Makros gibt es vordefinierte Strings, die man in den Datenbanken tunlichst benutzen sollte; sie heißen `section`, `paragraph`, `appendix` und `part` und sind gedacht für den Einsatz im `TYPE`-Feld mancher Einträge, mit welchem sich das `CHAPTER`-Feld umwidmen läßt.

Benutzungshinweise

Zum Glück braucht man die im letzten Kapitel vorgestellten Makros nicht selbst einzugeben, sie sind vielmehr bereits in der Datei `apreamble.tex` enthalten, die zum Lieferumfang des Pakets gehört. Sie stellt in Abhängigkeit vom `abfull`-Schalter ausgeschriebene oder abgekürzte Füllwörter zur Verfügung. Mehr noch, diese Datei enthält

neben den deutschen Füllwörtern auch äquivalente Definitionen in englischer Sprache. Die Auswahl zwischen den beiden Sprachversionen wird automatisch getroffen, und zwar nach folgenden Kriterien:

- Ist das Makropaket `german.sty` *nicht* geladen, wird die englische Version genommen.
- Wird dagegen festgestellt, daß `german.sty` eingelesen wurde, hängt die Sprachauswahl vom eingestellten `\language`-Wert ab:
 - Ist dieser gleich `\german` oder `\austrian`, wird die deutsche Version gewählt.
 - In allen anderen Fällen ist es wieder die englische.

Auf diese Weise sollten zumindest alle diejenigen zufriedengestellt sein, die entweder nur englische Dokumente verfassen oder unter Benutzung von `german.sty` sowohl englische als auch deutsche Texte schreiben. Natürlich steht es jedem frei, `apreamble.tex` um zusätzliche Sprachen zu erweitern oder die mittels `\abtype` vorgenommenen Formatierungen nach eigenem Gutdünken abzuändern.

Damit der oben vorgestellte Mechanismus zur Sprachauswahl funktioniert, muß die Datei `german.sty` – wenn überhaupt – *vor* `apreamble.tex` eingelesen werden. Dies sollte im Zusammenhang mit \LaTeX kein Problem darstellen, da `german` für gewöhnlich als Dokumentstiloption verwendet wird. Wie oben erwähnt läßt sich \BibTeX aber auch mit `plain-TeX` nutzen, wenn das Makropaket `btxmac.tex` eingebunden wird; hier ist dann sicherzustellen, daß der Befehl `\input german.sty` vor `\input apreamble` (bzw., wie im nächsten Absatz erläutert, vor `\bibliography{...}`) erscheint. Alle zur *adaptable family* gehörigen Makros sind übrigens so abgefaßt, daß sie sowohl mit \LaTeX als auch mit `plain-TeX` funktionieren; insbesondere vertragen sie sich auch mit dem `WEB`-System für strukturierte Dokumentation.

Weiterhin ist es natürlich erforderlich, daß `apreamble.tex` vor der eigentlichen Bibliographie eingelesen wird; auch dafür steht eine implizite Lösung bereit. Es ist ohnehin vorteilhaft, wenn man eine oder mehrere `bib`-Dateien mit allgemeinen Definitionen (hauptsächlich `@STRING`-Kommandos) anlegt und diese stets am Anfang der `\bibliography`-Liste aufzählt. Zum Lieferumfang der *adaptable family* gehören zwei solche Dateien, `jourfull.bib` und `jourabbr.bib`, die Namen für Zeitschriften – einmal in ausgeschriebener und zum anderen in abgekürzter Form – definieren, und zwar gerade die, welche in den Standard-Bibliographiestilen (der *plain family*) bereits vordefiniert sind². Beide Dateien laden `apreamble.tex` automatisch am Anfang der Bibliographie. Alles, was man zur Nutzung der anpaßbaren Stile – abgesehen vom eventuellen Laden von `btxmac` – innerhalb eines Dokuments zu tun hat, beschränkt sich somit auf zwei Kommandos, die typischerweise so aussehen:

```
\bibliography{jourfull,user1,user2,user3}\bibliographystyle{aplain}
```

Dabei sind `user1`, `user2` und `user3` irgendwelche Benutzerdatenbanken, `jourfull` kann natürlich auch durch `jourabbr` ersetzt werden (muß aber immer am Anfang der Liste stehen!) und statt `aplain` kann selbstverständlich ein beliebiger anderer anpaßbarer Stil verwendet werden. Es ist zu betonen, daß die Auswahl „ausgeschriebene oder abgekürzte Zeitschriftennamen“ nicht vom Bibliographiestil bestimmt wird, sondern durch Angabe von `jourfull` oder `jourabbr` erfolgt.

Außerdem definieren die beiden Dateien `jourfull.bib` und `jourabbr.bib` Strings namens `ifger`, `else` und `fi`, die es auf einfache Weise ermöglichen, innerhalb der Datenbanken bei einzelnen Feldern von der automatischen Sprachauswahl zu profitieren. Beispielsweise könnte damit das `NOTE`-Feld eines Eintrags etwa folgendes enthalten:

```
ifger # "Wird demn{"a}chst erscheinen." # else # "To be published." # fi
```

Wie man unschwer erkennt, handelt es sich um eine bedingte Anweisung: `ifger` sorgt dafür, daß der nachstehende Text ausgegeben wird, wenn die automatische Sprachauswahl sich für „deutsch“ entschieden hat, `else` leitet die Alternative für den gegenteiligen Fall ein und `fi` schließt die ganze Konstruktion ab. Der Operator `#` verkettet die einzelnen Teile zu einem von `TeX` interpretierbaren Gesamtstring.

Gelegentlich mag es vorkommen, daß man die neuen, auf die *adaptable family* zugeschnittenen Datenbanken mit einem der Standardstile verwenden will. Dies stößt auf Schwierigkeiten, da die anpaßbaren Stile einige zusätzliche Strings definieren, von denen man sicherlich Gebrauch gemacht hat (genauer: machen sollte). Es steht aber eine Datei `acompat.bib` zur Verfügung, die alle zusätzlichen Strings nachträglich definiert. Bindet

²Der Autor der *adaptable family* war der Meinung, daß die Auswahl dieser Zeitschriften zu speziell auf ein Fach bezogen ist, und hat die Definitionen daher in seinen Stildateien weggelassen. Unter Benutzung dieser beiden Dateien ist jedoch die volle Kompatibilität von Datenbanken, die für die Standardstile erstellt wurden, zu den anpaßbaren Stilen gewährleistet.

man diese noch vor `jourfull.bib` oder `jourabbr.bib` (die man für die oben beschriebenen Mechanismen weiterhin braucht) in die `\bibliography`-Liste ein, so steht einer Nutzung der Standardstile etwa in der Form

```
\bibliography{acompat,jourfull,user1,user2,user3}\bibliographystyle{plain}
```

nichts mehr im Wege. Natürlich funktioniert die automatische Sprachauswahl jetzt nicht mehr bei den Füllwörtern, sondern nur noch bei den „handgemachten“ `ifger`-Konstruktionen.

Abschließend noch eine Anmerkung zur Groß-/Kleinschreibung von Zeitschriftentiteln. Die Stile der *plain family* setzen solche stets in Kleinschreibung um; so wird z. B. aus

```
TITLE="A Horror Story about Integration Methods"
```

die Titelangabe „A horror story about integration methods“. Es wurde verschiedentlich die Meinung vertreten, daß ein deutscher Bibliographiestil diese Umsetzung abschalten sollte, da sie für deutsche Titel nicht adäquat ist. Der Autor der *adaptable family* schließt sich dieser Meinung nicht an (und hat die Umsetzung daher beibehalten): Groß-/Kleinschreibung von Zeitschriftentiteln ist nicht eine Frage der Sprache, in der das Dokument gesetzt ist, sondern eine Funktion der Sprache des Titels selbst. So sollte die Eingabe

```
TITLE="Graphische Sprachelemente in ALGOL 68"
```

immer zu der Ausgabe „Graphische Sprachelemente in ALGOL 68“ führen – auch in einem englischen Dokument; ebensowenig ist etwas dagegen einzuwenden, wenn im obigen Beispiel mit dem englischen Titel die Umsetzung auch in einem deutschen Dokument stattfindet.

Es ist also *bei den einzelnen Titelangaben* dafür zu sorgen, daß die Umsetzung ggf. verhindert wird. Dies ist aber – bei den Standard- wie bei den anpaßbaren Stilen – leicht möglich, wie im `BIBTEX`-Manual [4] beschrieben wird: Man schließe den gesamten Titel in ein zusätzliches Paar geschweifeter Klammern ein. Also bewirkt

```
TITLE="{Graphische Sprachelemente in ALGOL 68}"
```

genau das Gewünschte (die Gänsefüßchen zählen als äußeres Klammerpaar und könnten auch durch geschweifte Klammern ersetzt werden, die zusätzlichen Klammern sind aber trotzdem erforderlich). Statt dessen könnte man auch nur die großzuschreibenden Wörter oder Buchstaben klammern, z. B.

```
TITLE="{G}raphische {S}prachelemente in {ALGOL} 68"
```

Diese Technik wendet man manchmal auch in englischen Titeln an, etwa in

```
TITLE="Stiff {ODE} Solvers: A Review of Current and Coming Attractions"
```

so daß sich der Titel „Stiff ODE solvers: A review of current and coming attractions“ ergibt (das Wort nach dem Doppelpunkt bleibt ohne besondere Maßnahme großgeschrieben).

Literatur

- [1] DONALD E. KNUTH. „Fundamental Algorithms“, Band 1 aus „The Art of Computer Programming“. Addison-Wesley, Reading, Massachusetts, Zweite Auflage (1973).
- [2] HELMUT KOPKA. „`LATEX` – Eine Einführung“. Addison-Wesley, Bonn, Dritte Auflage (1991).
- [3] LESLIE LAMPORT. „`LATEX`: A Document Preparation System“. Addison-Wesley, Reading, Massachusetts (1986).
- [4] OREN PATASHNIK. „`BIBTEXing`“ (Januar 1988).
- [5] OREN PATASHNIK. „Designing `BIBTEX` Styles“ (Januar 1988).