

Bijlage 12

Toolbox

Maarten Gelderman
Vrije Universiteit
Amsterdam
mgelderman@econ.vu.nl

abstract

Er lopen in de wereld zo veel \TeX -gebruikers rond, dat het zelden zal voorkomen dat een normale gebruiker de eerste is die een probleem tegen komt. Meestal zal dit probleem ook al door iemand anders zijn opgelost en vaak staat de oplossing op de 4all \TeX , \TeX live of een andere CD-ROM. Indien daar niets te vinden is staat er vast wel een oplossing op CTAN¹ of één of andere Internetsite. Het probleem zit hem niet in de beschikbaarheid, maar in het opsporen van de oplossing. Artikelen in MAPS zijn daarvoor natuurlijk een belangrijke informatiebron. Sommige oplossingen zijn echter zo eenvoudig dat er nooit een artikel aan gewijd zal worden. Over een draadstriptang valt nu eenmaal niet al te veel te vertellen. In *Toolbox* zal niet alleen de draadstriptang worden besproken, maar ook het gebruik van lucifers om het plastic van het einde van het snoer af te branden. Literatuurverwijzingen vormen deze keer de hoofdmoot van het verhaal.

Hoe werkt dat ook al weer?

Eén van de meest tijdsbesparende functies van \LaTeX is wel de mogelijkheid om op gemakkelijke wijze literatuurlijsten bij artikelen en boeken te maken, die 'altijd' in overeenstemming zijn met de laatste versie van het artikel. Dit scheelt een hoop controlewerk (haal ik geen literatuur aan die niet op mijn literatuurlijst staat en staan er op mijn literatuurlijst geen overbodige verwijzingen) en een hoop typewerk. De titelbeschrijvingen zelf, kunnen immers keer op keer opnieuw gebruikt worden. Laten we nog even kort herhalen hoe dit ook al weer werkt. In het \LaTeX -document doen we het volgende:

```
Een belangrijke verwijzing is \cite{Doll188}.
In een later artikel wordt letterlijk de
opmerking gemaakt: 'een letterlijke opmerking'
\cite[p.~455]{Doll194}. Een opmerking die meer
impliciet is terug te vinden in andere
artikelen over dit onderwerp
\cite{Alavi83,Benbasat83,Doll195}.
\nocite{Gelderman98hicss} % Mezelf haal ik
                        % helemaal niet aan,
                        % maar zo kom ik
                        % lekker wel op de
                        % literatuurlijst terecht

\bibliographystyle{plain}
\bibliography{data}
```

Dit voorbeeld laat 'alle' mogelijkheden van het `\cite`-commando op een rijtje zien. Met `\cite{naam1[,naam2...[,naamn]}` Worden één of meer literatuurverwijzingen ingevoegd. Indien slechts één verwijzing wordt ingevoegd is het zinvol om bij voorbeeld een paginanummer mee te geven als optioneel argument. Het `\bibliographystyle`-commando geeft aan hoe de literatuurlijst eruit moet gaan zien (in dit geval is er gekozen voor de

1. CTAN is het Comprehensive \TeX Archive Network, de dichtsbijzijnde CTAN-site is <ftp://ftp.cs.ruu.nl/pub/tex-archive>. Ook <ftp://ftp.dante.de/pub/texarchive> is goed bereikbaar en biedt bovendien zoekfunctionaliteit.

standaardlijst `plain`, een genummerde en gesorteerde lijst. Als verwijzing worden in de tekst standaard de nummers tussen blokhaken (`'[1]'`) opgenomen. En het `\bibliography`-commando wordt gebruikt om aan te geven hoe het bestand met de benodigde gegevens heet. Het gebruik van meerdere bestanden is overigens ook mogelijk: net als bij het `\usepackage`-commando moeten de bestandsnamen gewoon door komma's gescheiden worden ingegeven.

Bij het de eerste keer runnen van \LaTeX genereert \LaTeX de foutmelding 'There were undefined references'. In de `.aux`-file die standaard door \LaTeX wordt aangemaakt zijn de literatuurverwijzingen (de `\cite`-commando's) echter wel opgenomen. Wanneer nu \BIBTeX wordt gedraaid leest dit de `.aux`-file in om allereerst vast te stellen welke database gelezen moet worden (in dit geval `data`) en welke stijl moet worden gebruikt voor de literatuurlijst (in dit geval `plain`). Vervolgens worden de literatuurverwijzingen ingelezen uit de database en opgemaakt conform de opgegeven stijl. De zo ontstane literatuurlijst wordt opgeslagen in een `.bbl` bestand. \LaTeX leest bij een volgende run dit bestand automatisch in, en na twee keer draaien (net als bij het gebruik van `\label` en `\ref` is twee keer runnen van \LaTeX noodzakelijk) is de mededeling over de ongedefinieerde verwijzingen verdwenen en is op de plaats van het `\bibliography`-commando de literatuurlijst toegevoegd.

Voordat dit werkt, moeten we natuurlijk wel een database hebben. Lamport [1] beschrijft in appendix B hoe zo'n database eruit dient te zien. De database is een ASCII-file en de titels dienen hier in te staan op een wijze die vergelijkbaar is met het volgende voorbeeld:

```
@STRING{MISQ = "Management Information Systems Quarterly"}

@ARTICLE{Doll194,
  author = {Doll, William J. and Xia, Weidong and Torkezadeh, Gholamreza},
  title = {A confirmatory factor analysis of the end-user computing
    satisfaction instrument},
  year = 1994,
  journal = MISQ,
  pages = {453-461},
  month = dec
}
```

Op de eerste regel wordt een afkorting gedefinieerd.² De overige regels bevatten de beschrijving voor een artikel, waarin onder andere de betreffende afkorting en de voorgedefinieerde afkorting `dec` worden gebruikt. Daar de database een gewone ASCII-file is, kunnen we deze in dezelfde editor onderhouden, als waarin we de \TeX -bestanden zelf ingeven. Er zijn echter weinig mensen die het intypen van dit soort data tot hun hobby rekenen en bovendien is de kans op fouten maar al te groot. Derhalve ga ik in de volgende paragraaf in op wat programma's die het bewerken van de database vergemakkelijken. Daarna kom ik toe aan mogelijkheden om de layout van de database en de literatuurlijst te veranderen.

Het bijhouden van de database

Diegenen die met emacs werken hebben weer eens geluk. Zij hebben vermoedelijk reeds lang geleden $\text{auc}\TeX$ geïnstalleerd. Wanneer een `.bib`-file wordt opgevraagd switcht emacs automatisch naar $\text{BIB}\TeX$ -mode. Via de menu's of met behulp van een paar toetsaanslagen kunnen nu de namen van de velden van de database automatisch worden

2. In de praktijk is het handig de afkortingen in een apart bestand te bewaren. Ook artikelenbundels en dergelijke plaats ik altijd in een aparte database, voor gebruik van de cross-ref functionaliteit moet $\text{BIB}\TeX$ deze items immers als laatste inlezen. Mijn `\bibliography`-commando is normaalgesproken dan ook `\bibliography{afkortingen,gegevens,artikelenbundels}`.

ingevoegd. Verder is het mogelijk items te sorteren en de syntax van de buffer te controleren (zijn alle haakjes wel netjes gesloten, staan de komma's wel op de juiste plaats). Zoals gebruikelijk bij GNU-software is bestandsomvang geen probleem. Alleen de syntax-highlighting geeft het op met wat grotere bestanden, maar de bestanden zelf hebben hier niet onder te leiden en alle andere functionaliteit blijft beschikbaar (althoewel de syntax-controle met grotere bestanden wel erg traag wordt).

Wie niet met emacs werkt, of niet houdt van het bijhouden van ASCII-databases met behulp van een editor zal zijn heil moeten zoeken bij een meer specialistisch programma. Voor MS-DOS- en MS-Windows-gebruikers is de siamese tweeling BIBDB en BIBEDIT beschikbaar.³ BIBDB is zowel verkrijgbaar voor MS-DOS als in een MS-Windows-versie en bevat een forse (62 pagina's) handleiding. Het pakket is uitgebreid configureerbaar, kan Tib, Refer en comma-delimited bestanden inlezen, kan sorteren, kan filteren en kan (zeer krachtig) zoeken. Het belangrijkste nadeel is dat het pakket een nogal stugge interface heeft, waarbij voor iedere handeling een nieuw venster wordt geopend. Problemen die eerdere versies hadden met het inlezen van grote bestanden, lijken met versie 2.0 verholpen. Mijn eigen literatuurdatabase (met iets meer dan 1000 items, in totaal ruim over de 10.000 regels groot) werd in ieder geval zonder problemen ingelezen en kon ook gewijzigd worden. BIBDB is beschikbaar op <http://pluto.mpi-hd.mpg.de/~doron/bibdb.html> en <http://www.tcisoft.com/bibdb.html>. Op het moment dat ik deze *Toolbox* schrijf is de CTAN versie niet up-to-date.

BIBEDIT bevat geen handleiding, en dat is niet nodig ook. De interface conformeert zich aan hetgeen in de MS-Windows-wereld gebruikelijk is en iedereen met een beetje computerervaring zou er in één keer mee moeten kunnen werken. De functionaliteit is beperkt gehouden. BIBEDIT is in principe alleen geschikt voor het bijhouden voor databases, maar dat is voor de meeste gebruikers meer dan genoeg. Vervelend is dat velden maximaal één regel lang kunnen zijn en het programma zich derhalve minder leent voor het maken van 'annotated bibliographies'. Ook worden Tabs niet goed ingelezen. Problematisch is echter dat het pakket een (onduidelijke) beperkte capaciteit heeft. Van mijn literatuurdatabase was (zonder waarschuwing) na inlezen, bijwerken en opslaan de laatste 100 KiloByte verdwenen. Desalniettemin is BIBEDIT voor de niet veeleisende gebruiker (250 entries werden in ieder geval zonder problemen verwerkt) door de superieure interface een beter pakket dan BIBDB. BIBEDIT is dankzij de auteur van BIBDB beschikbaar op <ftp://ftp.tcisoft.com/tcisoft/BibEdit/>.

Er zijn nog vele andere pakketten om databases bij te houden. De twee besproken tools dekken tezamen met $\text{auc}\text{T}_{\text{E}}\text{X}$ echter precies af wat een normale $\text{T}_{\text{E}}\text{X}$ -gebruiker nodig heeft. BIBEDIT is voor de niet veeleisende gebruiker, BIBDB voor wie met grotere bestanden wil werken, veel functionaliteit wil hebben en bereid is te leven met een stugge interface en $\text{auc}\text{T}_{\text{E}}\text{X}$ voor de emacs-verslaafde. Het enige wat eigenlijk ontbreekt is een grafische interface voor Unix- en MacIntosh-gebruikers. *xbibtex* schijnt een bruikbaar programma voor Unix-gebruikers te zijn, maar weigerde op mijn PC te compileren. Voor de MacIntosh bestaat een programma genaamd *hyperbibtex*. Beide laatstgenoemde programma's zijn op CTAN te vinden.

Naast de pure database-programma's is op CTAN nog ander bruikbaars te vinden. Met name *bibclean* en *bibtool* zijn de moeite van het vermelden waard.⁴ Beide programma's helpen de gebruiker bij extraheren van een nieuw *.bib*-bestand uit een groter bestand met in het nieuwe bestand alleen de titels die in een opgegeven *.aux*-file worden genoemd (handig bij het inzenden van een artikel voor een tijdschrift). Verder is de functionaliteit gericht op het vinden van fouten in bestanden, controle op dubbele entries, het sorteren van bestanden etc. *bibtool* bevat een handleiding van 63 pagina's die in 1997 voor het laatst is

3. In de documentatie van BIBDB wordt de lezer die het pakket te moeilijk vindt aangeraden BIBEDIT te gebruiken, terwijl de auteur van BIBEDIT veeleisende gebruikers aanraadt over te stappen op BIBDB.

4. Naamgeving is bij dit soort programma's een probleem, op CTAN zijn meerdere *bibcleans*, een *bibtool* en een *bibtools* etc. te vinden. Zoek bij het downloaden dus even door.



Figure 1. De MS-DOS- en MS-Windows-versie van BibDB.

bijgewerkt. bibclean is gedocumenteerd middels een TugBoat-artikel van de auteur (Nelson Beebe) en bevat tevens een uitgebreide man-page. Geen van beide tools heeft een probleem met mijn database, maar alleen bibclean attendeert me op het feit dat een aantal annotate-velden te lang is. Onder Unix compileren beide pakketten zonder problemen, maar alleen bibclean heeft een aparte `ibmpc`-directory bij de broncode zitten. Vermoedelijk is het laatste pakket dus het geschiktst voor de MS-DOS/MS-Windows-gebruiker, een voorgecompileerde versie is echter van geen van beide pakketten beschikbaar.

Afwijkende formaten voor literatuurlijst en verwijzingen

Bij het gebruik van BIB_{TEX} blijken drie vragen bij herhaling op te komen:

1. Ik wil een literatuurlijst per hoofdstuk
2. Ik wil dat mijn literatuurlijst er anders uitziet
3. Ik wil anders citeren, bij voorbeeld niet [1], maar Jansen (1986)

Voor de eerste wens is `chapterbib` beschikbaar. Dit is een stijlbestand, en wordt dus opgeroepen met `\usepackage{chapterbib}`. Als ieder hoofdstuk in een apart bestand staat en wordt opgeroepen met `\include` wordt automatisch bij ieder hoofdstuk een aparte literatuurlijst opgenomen. Alhoewel dit natuurlijk geen ideale oplossing is, is het in de meeste gevallen toereikend. De documentatie van deze tool is onderaan de betreffende style-file

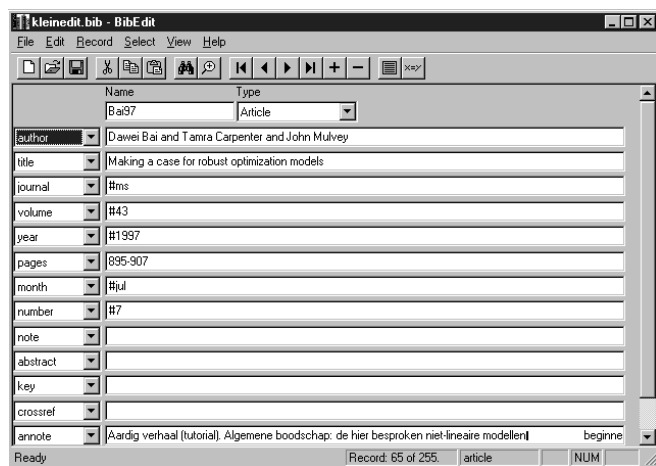


Figure 2. De MS-Windows-versie van BIBEDIT.

te vinden.

De tweede en derde vraag laten zich het beste gecombineerd beantwoorden. Wijzigingen van de wijze van verwijzen impliceren immers een ander soort literatuurlijst. In beide behoeften wordt voorzien door het package `natbib` en het plain- $\text{T}_{\text{E}}\text{X}$ -pakket `makebst` (beide tools zijn van de hand van Patrick Daly.).

Zoals hierboven reeds is aangegeven, dient in een $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -document waarin gewerkt wordt met literatuurverwijzingen die uit een database moeten worden gehaald het commando `\bibliographystyle` te worden gebruikt. De parameter van dit commando bevat de naam van een `.bst`-file die $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ inleest om uiteindelijk de bibliografie op te maken. Het grote probleem met `.bst`-bestanden is dat ze zo goed als onleesbaar zijn en de gemiddelde $\text{T}_{\text{E}}\text{X}$ -gebruiker is dan ook niet in staat om deze bestanden zelf aan zijn wensen aan te passen. Op CTAN is wel een grote hoeveelheid bestanden beschikbaar, maar deze doen over het algemeen precies niet wat je wilt, en bovendien is het nogal een klus om het juiste bestand te vinden. Dit probleem is met de komst van `makebst` grotendeels verdwenen. Na het intypen van `tex makebst` stelt $\text{T}_{\text{E}}\text{X}$ een schier eindeloze reeks met vragen over hoe de literatuurlijst eruit moet zien (waaronder de taal die gebruikt moet worden). Vervolgens kan het `.bst`-bestand automatisch gegenereerd worden. Door als argument van `\bibliographystyle` geen `plain`, maar de naam van het nieuwe `.bst` bestand te gebruiken, wordt *na de volgende* $\text{BIB}_{\text{T}}\text{E}_{\text{X}}\text{-run}$ de literatuurlijst opgemaakt conform de nieuwe eisen.

Het maatje van `makebst` is `natbib`. Dit is een gewone style-file, die (soms in combinatie met een door `makebst` aangemaakt `.bst`-bestand) vele nieuwe mogelijkheden biedt voor het opmaken van de literatuurverwijzingen zoals die in de tekst komen te staan. De functionaliteit van `natbib` is te uitgebreid om hier volledig te bespreken, wie het onderste uit de kan wil halen zal de handleiding die bij het package zit (20 pagina's) moeten lezen. Een korte bespreking moet hier volstaan.

Afhankelijk van de optie waarmee we `natbib` aanroepen, wordt het soort citaten aangepast. Als we `natbib` aanroepen met de `\usepackage[numbers]{natbib}` verandert er vrijwel niets, we houden gewoon de traditionele wijze van citeren. Met de extra opties `sort` en `sort&compress` wordt er bij meerdere verwijzingen echter zorg voor gedragen dat ze gesorteerd (en samengevoegd worden). `[1,4,7,3,2]` wordt met `sort [1,2,3,4,7]` en met `sort&compress [1-4,7]`. Met de optie `super` in plaats van `numbers` wordt er superscript gebruikt voor literatuurverwijzingen. `\usepackage[sort&compress,super]{natbib}` resulteert dan in de verwijzing $^{1-4,7}$. Interpunctie etc. kunnen naar wens worden aangepast.

Indrukwekkender worden de veranderingen wanneer we de optie `authoryear` gebruiken. Het is dan wel noodzakelijk met een door `makebst` gegenereerde `.bst`-file voor auteur-jaar citatie te werken (en op de vraag of dit bestand geschikt moet zijn voor `natbib` ‘ja’ te antwoorden). De meest elementaire voorbeelden van commando’s die dan mogelijk zijn kopieer ik direct uit de handleiding.

<code>\citet{jon90}</code>	⇒	Jones et al. (1990)
<code>\citet[chap.~2]{jon90}</code>	⇒	Jones et al. (1990, chap. 2)
<code>\citep{jon90}</code>	⇒	(Jones et al., 1990)
<code>\citep[chap.~2]{jon90}</code>	⇒	(Jones et al., 1990, chap. 2)
<code>\citep[see][]{jon90}</code>	⇒	(see Jones et al., 1990)
<code>\citep[see][chap.~2]{jon90}</code>	⇒	(see Jones et al., 1990, chap. 2)
<code>\citet*{jon90}</code>	⇒	Jones, Baker, and Williams (1990)
<code>\citep*{jon90}</code>	⇒	(Jones, Baker, and Williams, 1990)

Ook hier kan de interpunctie naar wens worden aangepast. Bovendien is het mogelijk automatisch auteursindexen te laten aanmaken.

Al die verwijzingen

Maar daarmee genoeg over literatuurverwijzingen. Nog slechts twee opmerkingen in deze *Toolbox* en wel over verwijzingen van een heel andere soort: URL’s. In het vorige nummer van *Toolbox* heb ik eindeloos lopen emmeren om die erin te krijgen, met name die tildes zijn zo vervelend. Taco had echter de oplossing: gewoon het package `url` gebruiken en het commando `\url` komt ter beschikking. Het gebruik spreekt voor zich.

Mocht in dit artikel de URL eens niet beschikbaar zijn, dan gewoon zoeken op CTAN. Dante is daarvoor geschikter dan Utrecht, het biedt een eenvoudige zoekfunctie: inloggen op de bekende manier en vervolgens `quote site index zoektekst` ingeven. Met `quote site index url` moet bovenstaand package te vinden zijn.⁵

Volgende keer

Geen idee, waar ik het de volgende keer over ga hebben. Misschien wel iets over conversie van en naar andere tekstverwerkers. Mijn email-adres staat boven deze bijdrage: suggesties voor andere onderwerpen en te bespreken tools en trucs zijn welkom.

References

- [1] Leslie Lamport. *LT_EX—A Document Preparation System—User’s Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.

5. Er wordt natuurlijk gezocht naar alle bestanden die het woord ‘url’ bevatten. De bedoelde file bevindt zich, zoals verwacht ergens in de directory `/macros/latex`.