

Aanwezig : A. Bakker; P.G.M. Bezembinder (Thieme); L. de Coninck (de Kraal); G. van den Dobbelsesteen (LOGIN BV); W. Dol; R. Doornebal (Kluwer); P. van der Ende (IRI); E. van Eynde (Kath. Universiteit Leuven); G. Ervynck (Kath. Universiteit Leuven); E. Frambach (RUG); J. de Gast (Scan Laser B.V.); G. Geens (TFCG/IMEC); F. Goddijn (FGBBS); J. Hagen (Pragma); H. Haverkort; R. van der Heijden (Hogeschool van Utrecht); A. Heijs (Staring Centrum); T. Hoekwater (Kluwer); B. Jackowski (GUST); P. de Jong; N.S. Kroonenberg (RUG); R. Kwee; C.G. van der Laan; A. de Leeuw van Weenen (RUL); P. Maziers (Groupe De Boeck S.A.); H. van der Meer (UvA); C.C.M. Moes (TUD); P. von Mouche (LUW); H. van Mourik (Ned. Historisch Data Archief); G.J.H. van Nes (ECN); G. Oomen (Kluwer); P. van Oostrum (UU); J. Pijnenburg (KUB); R. Rietman; E. de Rouville; A. van Ryckeghem (Kath. Hogeschool Brugge-Oostende); A.C.A. Serier; W. Smit; A. Soos (UT); R.C.W. Strijbos (UT); E. v.d. Storm; P. van Summeren; M.S.J. Tünnissen; P. Tutelaers (TUE); B. Verheghe (Univ. Gent); J.E. van Weerden (UU)

Notulist : Frans Goddijn

1 Opening

Voorzitter Erik Frambach opent de achttiende NTG-bijeenkomst en dankt gastheer Jules van Weerden. Een bijzonder woord van welkom is er voor Bogusław Jackowski uit Polen. Hij komt een lezing verzorgen, gevolgd door een cursus op vrijdag. *Last minute* aanmeldingen kunnen nog via Kees van der Laan.

2 Notulen van 17de NTG-bijeenkomst van 1 juni 1996

De notulen worden ongewijzigd vastgesteld.

3 Ingekomen stukken en mededelingen

Secretaris Gerard van Nes geeft een korte opsomming van de binnengekomen tijdschriften van diverse zusterverenigingen. Hij vertelt ook van het initiatief om met een NTG-delegatie aanwezig te zijn op de aankomende landelijke bijeenkomst van de nederlandse vereniging van Wiskundeleraren, op 16 november.

Jules van Weerden vraagt of het mogelijk is de leden voortaan ook met voornaam aan te duiden op de badges. Hierover wordt van gedachten gewisseld en Piet Tutelaers komt op het lumineuze idee om voortaan de naam te gebruiken (ook op de badges) zoals deze door het lid zelf wordt opgegeven.

Het boek van Salomon is momenteel in herdruk.

Van de TeTeX UNIX CD zijn nog twee exemplaren bij Erik Frambach in voorraad. De rest is bij UKTUG te bestellen.

Addison & Wesley uitgeverij is met een stand aanwezig en de leden kunnen tegen 10% korting boeken aanschaffen.

Kees van der Laan meldt dat er een onderwijsdag *interactieve media* aankomt, georganiseerd door het Instituut voor Toegepaste Wiskunde. Op TEX-NL zal Kees van der Laan hierover nader berichten.

4 Bestuursmutaties, penningmeesterschap

Erik Frambach licht toe, aan de hand van emails door Wietse Dol, hoe de noodzaak van overdracht van het penningmeesterschap is gebleken. Het bestuur (waarvan de penningmeester deel uitmaakt) stelt voor om Wietse Dol als penningmeester zijn taken te laten uitvoeren tot het einde van 1996. Op de voorjaarsbijeenkomst kan dan een nieuwe penningmeester worden benoemd. Er is een kandidaat (Wybo Dekker) van het bestuur, en er kunnen eventuele tegenkandidaten komen. De vergadering heeft geen bezwaar tegen deze manier van aanpak.

5 Subsidieaanvraag FGBBS

De vergadering is na een korte toelichting door Frans Goddijn akkoord met de aanvraag.

6 Begroting 1997

Hierbij waren geen opmerkingen

7 MAPS zaken (vacature hoofdredacteur)

Nu Wietse Dol zijn hoofdredacteurschap heeft neergelegd, zijn in de MAPS-redactie in elk geval over: Jos Winnink (jos.winnink@cpb.nl), Gerard van Nes en Gilbert van den Dobbelsteen (gilbert@login.iaf.nl) (naast Wietse Dol). Hoe kan het beste het vele MAPS-werk worden verdeeld, over mensen en door het jaar heen?

Kees van der Laan signaleert dat er een scheiding is te maken tussen het zuiver technische werk en het redactionele werk aan de MAPS. Tot nu toe, door grote pieken in het productiewerk, is er minder tijd voor redactionele overwegingen dan we eigenlijk zouden willen.

Erik Frambach vindt dat idealiter het werven en selecteren van kopij, en de productie, doorlopend en afzonderlijk plaatsvinden met slechts een kleine piek-belasting van de redactie ten tijde van de uiteindelijke productie van de MAPS. Siep Kroonenberg denkt dat dit misschien wel *meer* werk oplevert, maar Erik Frambach vindt dit minder een bezwaar omdat het kleine meerwerk over veel meer tijd kan worden verspreid. Siep vindt het ook een goed idee om de deadlines eerder te stellen.

Taco Hoekwater is beschikbaar voor productiewerk, en ook Roland Kwee (niet in adressenlijst 24 april) stelt zijn tijd en inzet hiervoor beschikbaar.

De MAPS redactie zelf wil graag het initiatief houden in de diverse noodzakelijke wijzigingen en zal over een maand komen met een eigen plan op basis van bovengenoemde overwegingen.

Aangezien de MAPS voor de NTG een kernactiviteit vormen, zijn goede plannen en afspraken van het grootste belang.

7.1 MAPS Award

De zoete *stretchables* gaan ditmaal naar Keith Reckdahl voor zijn artikel “using EPS graphics in L^AT_EX 2_ε documents”. Keith kon deze dag helaas niet aanwezig zijn voor de uitreiking, maar de prijs wordt hem nagezonden.

8 Komende activiteiten (zit de NTG wel op de goede weg?)

We willen graag weten hoe de MAPS wordt gewaardeerd en of de NTG wel op de goede weg is naar de toekomst van T_EX. Begin 1997 zal er daarom een enquête worden toegestuurd aan de leden. De resultaten zullen worden verwerkt met de hulp van de statistische kennis en programmatuur bij Hans Hagen en gepresenteerd met behulp van Herman Haverkort die al eerder een stylefile voor het verwerken van enquêtegegevens ontwikkelde.

8.1 Lustrumcommissie 1998

Hierin zullen de eerste leden Hans Hagen en Peter van Summeren (pjhamsvs@fwi.uva.nl) graag andere leden verwelkomen om samen aan een goede lustrumviering te

gaan werken. Piet Tutelaers kom met de suggestie om Eberhard Mattes (serieus) en/of Hippocrates Sendoukas (jisendo@leon.nrcps.ariadne-t.gr; (grapje) uit te nodigen als spreker.

9 Wat verder ter tafel komt & rondvraag

Jules van Weerden stelt voor dat we ntg.nl als domeinnaam gaan gebruiken. Adressen als maps@ntg.nl en ntg@ntg.nl worden zo mogelijk.

10 Verslag werkgroepen

Erik Frambach doet verslag van het resultaat van naspeuringen naar het al dan niet nog actief zijn van ooit ingestelde werkgroepen. Een aantal werd bemand door inmiddels uit het zicht verdwenen ex-leden en een aantal is nog wel bemand maar in diepe slaap.

Kees van der Laan zou de slapende groepen willen aanhouden en op afroep via TEX-NL weer actief willen laten worden. Hans Hagen stelt dat een werkgroep een officiële NTG-activiteit is met een doel en ook werkelijke *activiteit*. Slapende groepen zijn niet nuttig en niet nodig. Erik Frambach licht toe dat het *officieel* zijn van een NTG-werkgroep ook kan betekenen dat deze in aanmerking komt voor subsidie.

Gerard van Nes zou een structureel aanpak willen van de WERKGROEP EDUCATIE. Een aanbod van een jaarlijkse beginnerscursus en een jaarlijkse cursus voor gevorderden is dringend nodig. Frans Goddijn stelt dat zo'n actieve werkgroep veel kan betekenen voor de NTG en dat deze groep juist meer *de markt op* zou moeten gaan en zeker *niet* indutten.

Over deze werkgroep wordt nog verder nagedacht.

Voor de WERKGROEP SGML is van diverse kanten belangstelling, o.a. van John de Gast (jdgast@scanlaser.nl) (SGML-gg), Hans van Mourik, Piet van Oostrum, Peter de Jong (p.dejong@twi.tudelft.nl) en (aanmeldpunt) Taco Hoekwater.

Van de WERKGROEP WWW zijn geen actieve leden meer over. Het is zaak deze werkgroep nieuw leven in te blazen, aangezien we wel degelijk een Website hebben op www.ntg.nl

Voor de WERKGROEP NL-TEX melden zich Johannes Braams en Nico Poppelier (n.poppelier@elsevier.nl).

Namens de WERKGROEP SPELLING vertelt Piet Tutelaers over de totstandkoming van dit belangwekkende initiatief. Het werk aan nieuwe afbreek routines voor Nederlandstalige T_EX-gebruikers kreeg onverwacht een belangrijke uitbreiding door het werk aan een nieuwe, correcte, vrij verkrijgbare Nederlandse woordenlijst.

Voorzitter Erik Frambach sluit hierna de vergadering en hoopt iedereen weer te begroeten op de volgende bijeenkomst, vermoedelijk in Delft bij de TU.

Van de voorzitter

Erik Frambach

Het afgelopen half jaar is een relatief stille periode geweest. Dat is voor de afwisseling wel eens prettig. Niettemin is er aan het NTG-front en aan het \TeX -front een en ander te melden.

MAPS

De MAPS-redactie heeft een nieuwe hoofdredacteur: Taco Hoekwater heeft die taak overgenomen van Wietse Dol. Bedankt Wietse voor alle tijd en energie die jij er altijd ingestoken hebt, en succes Taco met deze zware taak.

Elders in deze MAPS kunt u lezen hoe sinds kort alle ooit in MAPS gepubliceerde artikelen in elektronische versie beschikbaar zijn gesteld. Overigens, nog steeds geldt dat de elektronische versie pas één jaar na het verschijnen wordt vrijgegeven, dus zeg uw lidmaatschap nog maar niet op, want dan loopt u met de MAPS een heel jaar achter.

Financiën

De NTG gaat een nieuwe penningmeester aanstellen. Op 1 januari jongstleden heeft Wietse Dol het penningmeesterschap officieel neergelegd en neemt Wybo Dekker het penningmeesterschap waar. Als er op de ledenvergadering geen tegenkandidaten zijn zal hij dus als nieuwe penningmeester worden aangesteld. Dank, Wietse, voor alle moeite, en success Wybo.

Op het ogenblik zit de NTG er financieel relatief warmpjes bij. Dat komt door de inkomsten uit de verkoop van de nog steeds goed lopende 4all \TeX cd-rom. Dat geld kan wellicht nuttig besteed worden voor de lustrum-bijeenkomst volgend jaar. We vierden dan onze 10^e verjaardag! Hans Hagen leidt de lustrum-commissie en kan nog best extra leden gebruiken. Wie goede ideeën heeft hoe de lustrum-bijeenkomst vorm gegeven kan worden wende zich tot Hans.

4all \TeX

Die inkomsten uit de verkoop van de 4all \TeX cd-rom gaan (helaas voor de NTG) vervallen. De auteurs van de cd-rom, Wietse Dol en ondergetekende, werken aan een nieuwe versie die door Addison-Wesley zal worden uitgegeven. Voor het bevorderen van het gebruik van \TeX (de hoofddoelstelling van de NTG) is dat echter een goede zaak. Via zo'n wereldwijd opererende uitgever kan een veel groter publiek bereikt worden. Althans in theorie. Hoe het werkelijk uitpakt zullen we moeten afwachten.

\TeX Live

Er wordt weer druk gewerkt aan een nieuwe editie van de befaamde ' \TeX Live' cd-rom. Sebastian Rahtz c.s. heeft er weer een plaatje gemaakt dat elke \TeX -gebruiker op Unix-systemen moet aanspreken.

Nieuw op deze cd-rom zijn de 'Web2c for Win32' binaries. Web2c is nu ook geport naar de Microsoft Windows 95 en Windows NT-omgeving. Fabrice Popineau, een enthousiaste ex-Linux-gebruiker is hiervoor verantwoordelijk. Dat betekent dat Web2c nu beschikbaar is op ongeveer alle Unix-varianten, VMS, MVS, Amiga, OS/2 en Win32. Ik denk dat de meeste deskundigen het er wel over eens zijn dat Web2c de beste implementatie is die er is. Nog even en Web2c draait echt overal (waar blijft de Mac?). Die stroomlijning heeft grote voordelen. Implementatieafhankelijke verschillen worden geminimaliseerd en onderhoudstaken kunnen beter verdeeld worden. We zijn dan niet meer afhankelijk van de inzet van een enkeling, maar kunnen *echt* samenwerken – is het u opgevallen hoeveel namen ik in dit artikel noem? Ik ga nog even door.

Karl Berry is de drijvende kracht achter Web2c. Dankzij Bernd Raichle zit e- \TeX (een initiatief onder leiding van Philip Taylor) er ook bij, en zelfs PDF \TeX (een initiatief van Petr Sojka) zit erbij! Jammer dat Piet Tutelaers' PS2PK nog steeds ontbreekt.

TDS

Met Web2c wordt het ook gemakkelijker om \TeX inputs (\TeX files, style files, TFM's, VF's, PK's etc.) uit te wisselen, althans als we allemaal conformeren aan de ' \TeX Directory Structure', kortweg TDS. Is 'conformeren' een vloek? Ik denk het niet, als alle betrokkenen er baat bij hebben en dat is zeker het geval bij het globaal uitwisselen van bestanden. Laten we onze creativiteit uitleven op de *inhoud* van bestanden, en niet meer zeuren over waar iets in een boom opgeslagen moet worden. De ' \TeX Working Group' die de TDS heeft bedacht bestaat uit een groot aantal verstandige mensen (o.a. Sebastian Rahtz, Norman Walsh, Jörg Knappen, David Carlisle en Alan Jeffrey) die hier hard over hebben nagedacht. TDS is de standaard, lang leve TDS!

Uiteraard zal de nieuwe 4all \TeX cd-rom die speciaal gericht zal zijn op MS-Windows-gebruikers ook op Web2c gebaseerd zijn. Jammer voor em \TeX , maar Eberhard Mat-

tes is kennelijk te zeer gehecht aan zijn eigen spullen en aan OS/2 om in deze stroom mee te gaan.

TUG'97

Het thema is dit jaar: "TeX Comes Home". Dat klinkt erg vaag, maar om toch een idee te geven waar het over zal gaan worden de volgende deelgebieden genoemd:

- Picture this: the TeXie approach to graphical illustration
- Tooling up: where are we with TeX?
- TeX and Scientific Publishing on the Internet
- TeX behind the scenes: what is our relationship to SGML?
- TeX and the real world
- L^ATeX — state of the art
- Multilingual typography without boundaries

Dat kan interessant worden. De conferentie wordt gehouden in San Francisco, wat voor vele Europeanen helaas te ver weg (lees: duur) is. Op de TUG WWW-site (<http://www.tug.org/tug97/>) kunt u er meer over lezen.

EuroTeX?

Iets dichterbij huis dan. Helaas, dit jaar komt er geen EuroTeX-conferentie. Door allerlei omstandigheden lukt dat niet. Maar dat betekent niet dat er niets gebeurt. De Polen houden traditiegetrouw hun Bachotek-conferentie en gaan steeds meer op de internationale tour door veel

buitenlanders uit te nodigen. Dit jaar hebben ze beloofd voor goede Engelstalige samenvattingen te zorgen zodat het ook voor niet-Polen goed te volgen zal zijn. Verder is de omgeving prachtig (midden in de bossen bij een groot meer) en is de prijs bijzonder laag. Dit is beslist een aanrader!

Verder schijnt UKTUG plannen te hebben om ergens in het najaar iets te organiseren waarvoor leden van andere LUGs ook worden uitgenodigd. We zullen het nog wel zien.

Of er dan volgend jaar weer een echte EuroTeX-conferentie komt? Mogelijk, of misschien weer gecombineerde TUG/EuroTeX-conferentie zoals vorig jaar in Dubna, Rusland. Er hebben zich al verschillende kandidaten gemeld, o.a. Polen (alweer, wat zijn ze actief daar!) en Spanje (Barcelona). Die laatste wordt vast een klapper als die in de (na)zomer valt. Plak er een weekje aan vast en je kunt het nuttige met het aangename combineren.

En verder...

...de ontwikkelingen blijven volgen en zien hoe TeX in deze steeds sneller veranderende wereld stand kan houden. Hopelijk kan de TUG'97-conferentie daaraan bijdragen. TeX moet ook met zijn tijd meegaan, in wat voor vorm dan ook. Misschien moeten we eens over JavaTeX gaan denken. WordPerfect schijnt al zover te zijn, en de hele Corel suite komt eraan. Ook Lotus komt binnenkort met zijn Java-versie. Waar blijft TeX? Wie mij de eerste 'TeX Beans' kan laten zien krijgt van mij een pak mokkakoffie en een mokkataart!

Redactioneel

Taco Hoekwater

Alweer een nieuwe hoofdredakteur

Het zijn rumoerige tijden voor de MAPS. Kort na de vervanging van Gerard van Nes door Wietse Dol, gaf ook Wietse aan te weinig tijd te hebben om zich voluit op de MAPS te kunnen storten. Tijdens de najaarsbijeenkomst van de NTG werd het duidelijk dat de MAPS-redactie ‘onthoofd’ was. Hoewel de discussie tijdens de vergadering een ruime verzameling goede ideeën opleverde, was er nauwelijks sprake van een nieuwe hoofdredakteur. Er bleef het bestuur uiteindelijk slechts een weg open: omkoping!

Een nieuwe hoofdredakteur lijkt in veel opzichten op een zojuist afgestudeerde student: veel geleerd, en niets van die kennis blijkt ook maar enigszins van toepassing te zijn op de werkelijkheid. Uiteraard overkwam mij dat ook. Het heeft maanden geduurd voordat ik er ook maar een beetje grip op kreeg. Alweer is elke lezer van de MAPS heel veel dank verschuldigd aan Gerard van Nes, die erg veel tijd heeft gestoken in het vergroten en verduidelijken van mijn beeld over de MAPS en haar redactie. Dit is ook het moment om even Kluwer Academic (mijn baas) te bedanken voor het feit dat ik een gedeelte van mijn werkuren mag besteden aan de productie van de MAPS.

Op dit moment bestaat de redactie van de MAPS uit slechts vier personen: Gilbert van den Dobbelsteen, Jos Winnink, *natuurlijk* Gerard van Nes, en ondergetekende. Te weinig. We hebben met zijn vieren ontzettend veel werk moeten verzetten om te zorgen dat deze MAPS toch nog min of meer op tijd klaar was. We zijn dan ook naarstig op zoek naar extra ondersteuning.

Veranderingen in MAPS #18

Deze MAPS is grotendeels ontstaan in een niemandsland, en ziet er daardoor nog niet helemaal uit zoals we dat graag zouden zien. Toch zijn er al een paar dingen veranderd. Het opvallendste daarvan is dat met ingang van deze MAPS de paginanummering veranderd is.

Naast de ‘minutes’ (verslag) en de ‘appendices’ (bijlagen) is nu een ‘extra’ soort artikel, het ‘bericht’. *Berichten* bevatten artikelen die vroeger gezien werden als bijlage, en die als kenmerk hebben dat ze een direct verband hebben met werkzaamheden van de NTG zelf, zoals jaar- en werkgroepsverslagen, de begroting, etc.

Daardoor werd het mogelijk de *bijlagen* van een andere pagina-nummering te voorzien. Het voordeel van het

nieuwe systeem is dat een bijlage nu niet meer uitdrukkelijk is verbonden met zijn plaats in de MAPS, waardoor het werk voor de volgende MAPS vereenvoudigd gaat worden. Er hoeft niet meer gewacht te worden met verwerking en plaatsing binnen de MAPS totdat alle artikelen binnen zijn, omdat het paginanummer niet meer nodig is. Dat betekent dat de weg open is voor verwerking *per artikel*, met als uiteindelijk gevolg (hopen we) minder tijdsdruk.

Een verandering achter de schermen: deze MAPS is *gedrukt* van 1200 dpi uitvoer, de hoogste resolutie tot nu toe voor de productie van de MAPS. Het verdwijnen van de vorige ‘drukker’ zorgde voor de noodzaak een andere oplossing te vinden, en daarbij hebben we natuurlijk meteen de kans gegrepen om het ook *beter* te gaan doen!

De bijdragen

De artikelen in deze MAPS vallen met name op door een wijde verscheidenheid van onderwerpen. De volgorde moet met een korreltje zout genomen worden, maar we hopen dat er in ieder geval voor een ieder wel wat interessants bij zal zitten.

Traditioneel volgt er nu een kort stukje over de meest opvallende artikelen, maar ik zou niet weten welk artikel er nu het meest uitspringt. Ik voorzie nog grote discussies bij het beslissen over de MAPS-award!

Eén bijzonder artikel was in ieder geval het verhaal over fonts, dat voor indrukwekkende logistieke problemen zorgde, zoals te zien is aan het aantal gebruikte fonts:

```
Here is how much of TeX's memory you used:
3470 strings out of 55918
32230 string characters out of 211187
102201 words of memory out of 308997
6151 multiletter control sequences out of 10000
90305 words of font info for 220 fonts,
                                out of 262141 for 254
21 hyphenation exceptions out of 607
```

Ik noem met opzet geen namen, en wil hier namens de redactie en de lezers alle auteurs van harte bedanken voor hun bijdragen, die zonder uitzondering allemaal leerzaam waren. Het werk voor de redactie zit er weer op, over een kwartier gaat deze MAPS naar de drukker, en wij kunnen weer even bijkomen en krachten opdoen voor het najaar.

Veel leesplezier, en op naar de volgende MAPS!

Het weten waard

T_EX kalender 1997/1998

- 28 juli – 1 augustus **TUG '97**
San Francisco, California, USA
- 25 – 26 september **4^{de} Int. Conf. Hypertext & Hypermedia**
Parijs, Frankrijk
- 9 – 10 oktober **Bijeenkomst Dante**
Zeuthen (bij Berlijn), Duitsland
- maart 1998 **EuroT_EX '98**
Saint Malo, Frankrijk

1 Glossary

Gebruikersgroepen

- TUG : T_EX Users Group
LUG : Local Users Group
CSTUG : LUG Tsjecho Slowakije
CyrTUG : LUG USSR
DANTE : LUG Duitsland
GUTenberg : LUG Frankrijk
HunTUG : LUG Hongarije
ITALIC : LUG Ierland
JTUG : LUG Japan
Nordic : LUG Scandinavië, Denemarken, en IJsland
NTG : LUG Nederland en België
SibTUG : LUG Siberië
UKTUG : LUG Engeland
YUNUS : LUG Turkije (feitelijk een discussielijst)
GUST : LUG Polen

Bulletins/journals

- Baskerville : UKTUG
Cahiers GUTenberg : GUTenberg
Zpravodaj : CSTUG
TeXnische Komödie : DANTE
TeXline : Malcolm Clark; UK
GUST bulletin : GUST
TTN : T_EX and TUG News;
TUG
TUGboat : TUG
MAPS : Minutes and
Appendices; NTG

Diversen

- CTAN : Comprehensive T_EX Archive Network; sites waar men 'anonymous ftp' kan gebruiken om T_EX/L^AT_EX-achtig materiaal te verkrijgen. CTAN is de 'home' voor de officiële versie van L^AT_EX etc. CTAN sites zijn: ftp.dante.de, ftp.tex.ac.uk en pip.shsu.edu
- FGBBS : NTG's Bulletin Board
- AllT_EX : T_EX, L^AT_EX, etcT_EX
- ltxiii : L^AT_EX 3.0
- 4T_EX : Het volledige T_EX *runtime systeem* voor MS-DOS PC systeem, gebaseerd op emT_EX en 4DOS.
- 4allT_EX : De 4T_EX applicatie *plus* alle mogelijke gerelateerde files en utilities, gedistribueerd op de diskette set en CD-ROM.
- AMS : American Mathematical Society
- SGML : Standard Generalized Markup Language

2 NTG's T_EX Bulletin Board System

Op het T_EX Bulletin Board van de Nederlandstalige T_EX Gebruikersgroep (FGBBS) is een zo volledig en actueel mogelijke T_EX, emT_EX, L^AT_EX, TEX-NL en MusicT_EX collectie beschikbaar voor alle bezitters van een modem. Het BBS is kosteloos toegankelijk voor iedereen en er zijn geen beperkingen aan de hoeveelheid bestanden die kunnen worden opgevraagd. Het systeem is aangesloten op een High Speed modem, vergeleken met de transmissiesnelheid die een directe Internet link biedt misschien niet geweldig, maar veel beter kan het niet over de gewone huis- tuin- en keukenPTTlijn. De beheerders zijn Frans Goddijn en Henk de Haan. FGBBS is te bellen op 085-217041.

3 NTG's winkel

Via de NTG is beschikbaar:

- **4allT_EX dubbel CD-ROM:**
Ruim 1.1 Gbyte (ruim 50.000 files) aan onder meer 4T_EX utilities, fonts, T_EX/L^AT_EX/METAFONT/etc documentatie, de volledige MAPS 1 t/m 14, discussielijsten TEX-NL, TEX-HAX, UKTEX van de afgelopen 6 jaren, etcetc.
Kosten inclusief het uitgebreide 4T_EX manual: f 60,-.

- **T_EXlive CD-ROM:**

Kant-en-klaar T_EX systeem voor Linux, DEC, HP, IBM, NeXT, SGI, en SUN Unix systemen, alsook voor win32. Door Thomas Esser en Sebastian Raetz, gebaseerd op Web2C van Karl Berry.

Kosten inclusief manual: f 35,- (f 60,- voor niet-leden).

- **Syllabus Advanced T_EX course:**

Insights and Hindsights, David Salomon (*revised*; ruim 500 pagina's).

MAPS'92 speciale uitgave.

Kosten f 60,- voor leden en f 70,- voor niet-leden (extra verzendkosten: f 15,-).

- **NTG PR set**

Enkele mogelijkheden van (L^A)T_EX (ruim 25 pagina's). MAPS'93 speciale uitgave.

1 exemplaar gratis voor leden; extra exemplaren: f 7,50; niet-leden: f 15,- (inclusief verzendkosten).

Bestellingen kunnen gedaan worden door overmaking van het verschuldigd bedrag (plus verzendkosten) op de postgiro van NTG (1306238) t.n.v. penningmeester NTG, Eindhoven, met vermelding van hetgeen gewenst is.

4 NTG/TUG lidmaatschap

Het blijkt soms dat nieuwe NTG/TUG leden na ongeveer een half jaar nog geen TUGboat of TTN van TUG hebben ontvangen. Ondanks dat men een TUG lidmaatschap via NTG aanvraagt, blijkt in bijna alle gevallen de administratie- en verzendproblemen bij TUG zelf te liggen.

Mocht na enige maanden tijd geen post van TUG ontvangen worden, dan worden de betreffende NTG/TUG leden dringend verzocht om contact op te nemen met het secretariaat van de NTG.

5 De NTG voorjaarsbijeenkomst

De NTG voorjaarsbijeenkomst 1997 zal op donderdag 12 juni op de Technische Universiteit Delft plaatsvinden. Het formele gedeelte van de vergadering zal ongeveer een uur in beslag nemen.

6 MAPS 97.2

Sluitingsdata voor het inleveren van artikelen, bijlagen, en/of mededelingen voor de volgende MAPS uitgaven zijn:

1 oktober '97 (MAPS 97.2; #19)

1 april '98 (MAPS 98.1; #20).

Aanleveren kopij voor de komende MAPS:

- *Bij voorkeur* in L^AT_EX_{2 ϵ} gebruikmakend van de `maps.cls`. Deze stijlfile is via de redactie te verkrijgen en beschikbaar op de TEX-NL fileservers, `archive.cs.ruu.nl` (ftp-site) en FGBBS (085-217041).
- Daarnaast kunnen bijdragen ingestuurd worden gemaakt met `ltugboat.sty` of `article.sty` / `report.sty`.
- Verder zijn bijdragen vanzelfsprekend ook welkom in *plain-T_EX* of ongeformatteerd.
- Plaatjes bij voorkeur als (Encapsulated) PostScript file.

Daar MAPS bijdragen in plain T_EX altijd worden omgezet naar L^AT_EX, verdient vanzelfsprekend aanbidding van materiaal in L^AT_EX de voorkeur!

Eventuele nadere richtlijnen voor auteurs zijn op te vragen bij de redactie.

Bijdragen kunnen gestuurd worden naar:

`taco.hoekwater@wkap.nl`

Niet Internet-gebruikers kunnen hun bijdrage ook via modem/PTT lijn direct naar de redactie sturen. Gaarne hiervoor eerst contact opnemen met Taco Hoekwater.

Financieel verslag 1996

Erik Frambach & Wybo Dekker

1 Inleiding

Voor u ligt het financieel jaarverslag van de Nederlandstalige \TeX Gebruikersgroep over het jaar 1996. Dit verslag is opgesteld door Wietse Dol (penningmeester tot medio 1996) en Wybo Dekker (beoogde nieuwe penningmeester) en valt onder de verantwoordelijkheid van het NTG-bestuur.

De financiële ontwikkeling van de vereniging wordt aan de hand van de staat van baten en lasten en de balans besproken. Beide zijn in de tabel opgenomen, samen met respectievelijk de begroting en de overeenkomstige gegevens van een jaar geleden om zo vergelijking mogelijk te maken.

2 Baten

Contributies De contributies waren ruim begroot en zijn derhalve iets tegengevallen. Omdat het merendeel van de contributie besteed wordt aan MAPSen en dergelijke hebben de tegenvallende inkomsten geen groot (negatief) effect op het netto resultaat van 1996.

Rente De rente is aanzienlijk lager dan begroot. Dit komt omdat ervan uitgegaan is dat een aanzienlijk bedrag van de 4all \TeX CD-rom op de NTG rekening zou staan (en dus rente vangen). De rente wordt in maart uitgekeerd en voor de rente in deze afrekening gaat over het de rente van het jaar 1995. In 1995 werd euro \TeX gehouden. De hoge kosten daarvan leidden tot lage saldi op de NTG-rekeningen.

Verkoop Diverse oude en nieuwe MAPSen (specials) werden verkocht. Door de verkoop van de Unix \TeX -live CD is een aardige inkomsten bron ontstaan. Voor de opbrengsten van de 4all \TeX -CD bestaat de afspraak dat de NTG van de daarop gemaakte winst 50% zal ontvangen. Om vooral snog onduidelijke redenen zijn deze inkomsten tot heden buiten de boeken gebleven. In de loop van dit jaar zal daarover meer duidelijkheid ontstaan, van daar dat deze inkomsten als een pro memorie post zijn opgenomen.

3 Lasten

Administratie Kosten die gemaakt worden door de penningmeester en secretaris voor het versturen van allerlei verenigingspost en antwoorden op verzoeken om inlichtingen van potentiële nieuwe leden. Dit bedrag valt wel lager uit dan begroot

maar dat komt voornamelijk omdat veel post in combinatie met MAPSen kon worden verstuurd.

Afschrijvingen De NTG heeft een PostScript printer die in 3 jaar wordt afgeschreven. De laatste afschrijving vond in 1996 plaats zodat deze printer niet meer op de balans voorkomt.

Bestuurskosten Zo af en toe vergadert het NTG-bestuur om een aantal zaken die niet via e-mail afgehandeld kunnen worden te bespreken (dit vergaderen is soms telefonisch maar soms ook op locatie). Ook staan hier kosten die het bestuur maakt in verband met contacten met andere usergroups.

Crediteuren Deze post bestaat voornamelijk uit TUG-contributies (f 726) en stortingen die gedaan zijn door bestellers van de 4all \TeX -cdrom in 1995 (f 7165) en 1996 (f 2747).

KvK en notaris Jaarlijkse kosten en de kosten van een extra uittreksel als gevolg van de bestuurswisselingen.

Mededelingen aan de leden Hieronder valt de MAPS maar ook andere post. Uit de begroting blijkt dat we minder hebben uitgegeven dan begroot. Dit is het directe gevolg van het aantal leden dat lager was dan begroot. Ook is dit jaar geen extra MAPS special uitgegeven en waren de MAPSen minder dik dan de jaren daarvoor.

Reisbijdragen Dit jaar zijn alle reisbijdragen betaald door 4all \TeX of door de werkgever van bestuursleden.

Representatie Dit jaar is veel geld uitgegeven aan representatie en \TeX -promotie. Bij diverse bijeenkomsten waren NTG-bestuurders vertegenwoordigd. Als hoogtepunt is zeker de introductiedag voor wiskundeleraren te noemen, waaraan 20 leraren deelnamen.

Saldo Het positief saldo voor 1996 was f 3148.

Sponsoring In 1996 is f 1000 gegeven aan het FGBBS. Het bestuur heeft f 2015 geschonken als Bursary Fund voor de TUG-conferentie in Rusland. Het Bursary Fund stelde minder draagkrachtigen in staat aan de TUG-conferentie deel te nemen.

4 Conclusie

Gezien alle activiteiten en inzet van vele NTG-leden kunnen we dit jaar met een tevreden gevoel afsluiten. We moeten echter niet vergeten dat het positief saldo grotendeels het gevolg is van een bezuiniging op MAPSen en het niet declareren van reiskosten.

Baten en lasten over 1996

(bedragen in guldens)

	Begroting		Realisatie	
	Credit	Debet	Credit	Debet
Contributie	27 000		23 108	
Rente	1 400		404	
Verkoop TeX-live CD			914	
Verkoop 4allTeX CD			PM	
Administratie		1 000		480
Afschrijving		1 182		1 182
Bestuurskosten		750		685
Evenementen				565
Kamer van Koophandel		61		82
Mededelingen a/d leden		20 000		13 710
Onvoorzien		207		
Reisbijdragen		2 000		
Representatie		200		1 560
Sponsoring		2 000		3 015
Saldo				3 148
	28 400	28 400	24 426	24 426

Balans

	per 1-1-1996		per 12-1-1997	
	Activa	Passiva	Activa	Passiva
Giro	17 295		27 348	
Kas	100		100	
Debiteuren	80		80	
Apparatuur	3 546			
Contributies	1 690	715		413
Crediteuren		7 407		10 743
Afschrijving		2 364		
Kapitaal		12 225		16 372
	22 711	22 711	31 074	31 074

Jaarverslag NTG

jan–dec 1996

Gerard van Nes
secretaris

1 Algemeen

Het jaar 1996 was voor de NTG een druk en wederom succesvol jaar. Toppers waren onder meer de bijeenkomst met Knuth, de presentatie (plus cursus) aan de Wiskunde-docenten en de cursus voor NTG-leden door de Metapost/Metafont guru Bogusław Jackowski.

Twee NTG-bijeenkomsten (beide druk bezocht) vonden plaats met vele goede lezingen van sprekers van zowel binnen als buiten de NTG. Er verscheen wederom een tweetal uitgebreide uitgaven van de MAPS (Minutes & Appendices).

Natuurlijk zijn ook weer diverse leden actief geweest, bij onder meer het beheer van het Bulletin Board FGBBS, het beheer van TEX-NL, de uitgebreide hulpverlening bij vragen op de nationale TEX-NL en de ondersteuning van de internationale 4TEX-discussielijst.

2 Het NTG-bestuur

Het NTG-bestuur bestond in 1996 uit de volgende personen:

- E.H.M. Frambach, voorzitter,
- G.J.H. van Nes, secretaris,
- W. Dol, penningmeester (tot oktober 1996),
- J.L. Braams (tot juni 1996),
- F. Goddijn, bestuurslid,
- T.C.A.J. Hoekwater, bestuurslid (vanaf juni 1996),
- J. Hagen, bestuurslid (vanaf juni 1996).

Op de NTG-bijeenkomst in juni vonden de bestuursverkiezingen plaats. Wietse Dol en Frans Goddijn traden reglementair af. Beiden stelden zich echter herkiesbaar en werden vervolgens bij acclamatie herkozen. Johannes Braams trad daarnaast af wegens tijdgebrek. Als nieuwe bestuursleden werden Taco Hoekwater en Hans Hagen door het bestuur voorgedragen en vervolgens eveneens bij acclamatie gekozen.

Vanwege onder meer drukke werkzaamheden bij zijn werkgever, trok Wietse Dol zich op de najaarsbijeenkomst terug als penningmeester en bestuurslid. De penningmeesterschapszaken werden eind 1996 overgedragen aan Wybo Dekker, die op de voorjaarsbijeenkomst van 1997 wordt voorgedragen als bestuurslid en nieuwe penningmeester.

Wietse Dol rondde de penningmeesterswerkzaamheden over 1996 geheel af. De voorzitter nam vanaf oktober de

verantwoordelijkheden van de aftredende penningmeester op zich.

Philippe Vanoverbeke continueerde ook in 1996 zijn functie van NTG België-commissaris en zorgde daarbij voor optimale contacten met onze Vlaamse leden.

3 Het NTG-ledenbestand

De stijging die het ledenaantal vanaf haar ontstaan kende bleek in 1996 voor het eerst tot stilstand gekomen te zijn. Het aantal instituutleden bleef stabiel.

Eind	1990	1991	1992	1993	1994	1995	1996
Leden	117	159	193	232	265	286	285
Inst. leden	21	28	30	33	36	33	32

De contributie werd in 1996 niet verhoogd. De contributie-inning bleek — evenals in voorgaand jaar — weer de nodige inspanningen te kosten. Met ingang van de recente inning van de penningen is deze administratie dan ook door Wybo Dekker geoptimaliseerd.

De samenwerking van NTG met TUG — waarbij het lidmaatschap gecombineerd kon worden overgemaakt — bleek te resulteren in 81 NTG/TUG-leden eind 1996 (eind 1995: 85).

De snelheid van toezending van het TUG-tijdschrift 'TUGboat' werd zeer sterk verbeterd, doch in enkele gevallen deden zich nog problemen voor.

In de voorjaars-MAPS werd de volledige NTG-ledenlijst gepubliceerd.

4 Bijeenkomst Knuth met de NTG

Op 13 maart organiseerde de NTG een bijeenkomst met Donald Knuth in de Rode Hoed te Amsterdam. Ongeveer 35 (voor het merendeel) NTG-leden waren daarbij aanwezig. Knuth ging uitgebreid in op de vragen die de aanwezigen aan hem stelden.

Een letterlijk verslag van deze bijeenkomst is opgenomen in MAPS #16. De gehele bijeenkomst werd tevens op video vastgelegd.

5 De NTG-bijeenkomsten

Er zijn in 1996 ook weer twee NTG-bijeenkomsten georganiseerd welke gekenmerkt werden door een levendige discussie en een hoge mate van informatie-uitwisseling.

1. Op zaterdag (!) 1 juni 1996 bij de Hogeschool van Amsterdam.
Aanwezig waren 53 leden.
2. Op donderdag 24 oktober 1996 bij de Universiteit van Utrecht (Faculteit Letteren).
Aanwezig waren 46 leden.

De eerstgenoemde bijeenkomst, welke nu met veel belangstelling in een weekend plaats vond, had als thema: ‘ \LaTeX / \TeX in het Middelbaar- en Hoger Onderwijs’.

De volgende lezingen werden gehouden:

- *MS-Windows en $4\TeX$: de weg die voor ons ligt* door E.H.M. Frambach (Rijksuniversiteit Groningen) en W. Dol;
- *Gebruik \LaTeX bij Thieme t.b.v. het tweede fase onderwijs* door P.G.M. Bezembinder (Thieme, Zutphen);
- *Enkele ervaringen met \LaTeX* door R.B.J. Pijlgroms (Hogeschool van Amsterdam);
- *Het maken van een Natuurkundeboek en Natuurkundetoetsen m.b.v. \LaTeX* door C.M. Fortuin (Hogeschool Gelderland, Arnhem);
- *\LaTeX als hulpmiddel t.b.v. het Wiskundeonderwijs* door B.J. Geels (Adriaan Roland Holst school, Bergen NH);
- *Een eenvoudige \LaTeX proefwerkstijl en logging/analyse van $4\TeX$ -gebruik* door E.H.M. Frambach (Rijksuniversiteit Groningen);
- *(\LaTeX) in het onderwijs* door T.A. Jurriens (Rijksuniversiteit Groningen);
- *(Waarom) moeten studenten \LaTeX leren en wanneer?* door A.W.J. van der Meer (Universiteit Twente);
- *Een \LaTeX -cursus in Groningen* door R. Smedinga (Rijksuniversiteit Groningen);
- *\LaTeX onderwijs voor studenten* door P. Tutelaers (Technische Universiteit Eindhoven);
- *Het ontwikkelen, onderhouden en vormgeven van opleidingsmateriaal met behulp van $Con\TeX$ t* door J. Hagen (Pragma, Zwolle);
- *Het in kleur zetten van \TeX -sources* door J. Hagen (Pragma, Zwolle);

Op de najaarsbijeenkomst, welke als thema had ‘ \LaTeX / \TeX en Graphics’, vonden de volgende lezingen plaats:

- *Het raadsel van \TeX* door J. Hagen (Pragma, Zwolle);
- *Meta-Font or Meta-Graph* door Bugusław Jackowski (GUST, Polen);
- *Commerciële Graphics in \TeX* door T.C.A.J. Hoekwater (Kluwer, Dordrecht)
- *Blue.tex and its graphics* door C.G. van der Laan
- *Just a little bit of PostScript* door C.G. van der Laan

Van bovengenoemde twee vergaderingen (en lezingen) verschenen verslagen met (uitgebreide) bijlagen in MAPS #16 en MAPS #17. Enkele lezingen werden gepresenteerd door sprekers die door de NTG speciaal voor deze gele-

genheden waren uitgenodigd.

De leestafels met ‘ \TeX -gerelateerde’ boeken, eerdere uitgaven van de MAPS, diverse LUG- en andere tijdschriften, brochures en diverse aan \TeX verwante geschriften, trok zoals altijd veel belangstelling. Bij zowel de voorjaars- als de najaarsbijeenkomst konden de aanwezigen wederom Addison-Wesley (\LaTeX)/ \TeX /PostScript boeken met korting aanschaffen.

6 NTG-cursussen

Er werden in 1996 door de NTG twee cursussen georganiseerd:

- **Cursus Graphics & Any \TeX**

Op 25 oktober organiseerde de NTG voor ongeveer 40 deelnemers een cursus \TeX & Graphics met in het bijzonder Metafont/Metapost.

Docent was de GUST Metapost/Metafont guru Bogusław Jackowski. Kees van der Laan organiseerde de cursus van begin tot eind.

NTG-lid Rene van der Heijden verleende onder meer onderdak aan de buitenlandse gast.

- **Cursus voor wiskundeleraren**

Op zaterdag 21 december werd aan een 20-tal wiskundeleraren een eendaagse $4\text{all}\TeX$ -cursus door Wietse Dol en Erik Frambach bij de Universiteit Utrecht (gastheer Piet van Oostrum) geven.

7 De NTG-MAPS

Ook in 1996 verschenen er twee MAPS uitgaven met totaal ongeveer 330 goed gevulde bladzijden. Daarbij onder meer als inhoud: verslagen van nationale en internationale bijeenkomsten, artikelen m.b.t. de NTG-lezingen, zeer veel tutorial- en ander educatiemateriaal, \TeX / \LaTeX & PostScript bijdragen, font-, \TeX - en PC-artikelen, bijdragen m.b.t. \TeX en \LaTeX macro's, en algemene plus technische \TeX - en \LaTeX -bijdragen.

De MAPS-eindredactie was in 1996 voor het eerst sinds 8 jaren in andere handen: namelijk van Wietse Dol. Redactionele medewerking verleende Gilbert van den Dobbelseen, Gerard van Nes, Christina Thiele en Jos Winnink. Eind 1996 legde Wietse Dol het eindredactieschap neer. Met ingang van MAPS 18 (97.1) treedt Taco Hoekwater op als hoofdredacteur.

In 1996 verschenen geen MAPS-specials. Wel werd — mede i.v.m. de presentatie van de NTG aan de Nederlandse Vereniging van Wiskundedocenten — de NTG PR-set door Wietse Dol grondig herzien en in een handig A5-boekje uitgegeven.

MAPS-Awards werden in 1996 uitgereikt aan Christina Thiele (‘Knuth meets NTG members’; MAPS #16), Kees van der Laan (gehele oeuvre) en Keith Reckdahl (‘Using EPS Graphics in \LaTeX 2e Documents’; MAPS #17).

8 De NTG-werkgroepen

Bij een inventarisatie bleken in 1996 ook de laatste NTG-werkgroepen van het eerste uur te zijn ingeslapen. Activiteiten kwamen hoofdzakelijk van individuele leden. De veel gebruikte TEX-NL discussielijst zorgt reeds dikwijls voor een vorm van 'overleg'. Tijdens de najaarsbijeenkomst zijn daarom de slapende werkgroepen opgeheven.

9 De NTG 4all \TeX CD-ROM

De belangstelling voor de 4all \TeX CD-ROM bleef ook in 1996 bestaan. Op PC-gebied is de CD-ROM inmiddels internationaal een standaard.

In juli werden 500 extra exemplaren van de derde editie aangemaakt.

Veel werk werd in het gehele CD-ROM ontwikkel- en productieproces verzet door het koppel Wietse Dol & Erik Frambach. Diverse leden van het NTG-bestuur waren betrokken op het logistieke- en voorlichtingsvlak.

Weinig vooruitgang werd echter geboekt m.b.t. de onderhandelingen met de beoogde 4all \TeX uitgever Addison-Wesley.

10 NTG op Internet

De NTG was ook in 1996 op het WWW te vinden. Eind 1996 werd een begin gemaakt met een grondige herziening van alle NTG WWW-pagina's (realisatie april 1997).

De NTG WWW-site kreeg een definitieve plaats bij de Universiteit Utrecht. Jules van Weerden verzette veel werk voor de realisatie daarvan.

Het adres is: <http://www.ntg.nl/>

Door de NTG werd een speciale harde schijf aan de Universiteit Utrecht beschikbaar gesteld t.b.v. onder meer een volledig CTAN-mirror archief (<ftp://ftp.ntg.nl/pub/tex-archive/>). Piet van Oostrum stak veel tijd in deze werkzaamheden.

11 Verdere Bestuursactiviteiten

De NTG wisselde informatie uit met de LUGs: Exemplaren van de MAPS werden verstuurd naar de meeste zusterorganisaties (inclusief TUG). Van de Duitse en Franse gebruikersgroep ontving de NTG de periodieken (zie ook de leestafel tijdens NTG-bijeenkomsten!).

Mede namens de NTG werden in 1996 de volgende bijeenkomsten bezocht:

- **Bacho \TeX '96 te Polen.**

Van 2 tot 4 mei werd door Erik Frambach, Wietse Dol en Kees van der Laan deze Oost-Europese GUST'96 bijeenkomst bijgewoond.

- **TUG/Euro \TeX '96 te Dubna (Sovjet Unie).**

Van 28 juli tot 2 augustus.

Aanwezig Erik Frambach, Taco Hoekwater en Kees van der Laan.

Een verslag van deze bijeenkomst is in MAPS #17 (blz. 9-10) opgenomen.

- **Jaarvergadering Nederlandse vereniging voor Wiskundeleraren.**

Op zaterdag 16 november te Bilthoven.

In een zeer professionele stand werd de NTG en \TeX bij de groep van wiskundedocenten gepresenteerd.

NTG- \TeX standhouders waren Paul Bezembinder, Wietse Dol, Erik Frambach en Gerard van Nes

Op 13 maart en 2 oktober vonden bestuursvergaderingen plaats waarin naast de algemene gang van zaken binnen de NTG, onder meer de organisatie van de NTG-bijeenkomsten, Public Relation activiteiten, financiële zaken, de toekomst van \TeX en de NTG, CD-ROM activiteiten, instandhouding NTG-werkgroepen, cursusorganisatie, opvolging penningmeesterschap en MAPS-hoofredacteurschap, taakverdeling binnen het bestuur en de gebeurtenissen binnen TUG en andere zusterorganisaties ter sprake kwamen.

De NTG-secretaris publiceerde in het blad 'Euclides' van de Nederlandse Vereniging van Wiskundeleraren een artikeltje over \TeX en de NTG, hetgeen resulteerde in de aanwezigheid van de NTG op hun jaarvergadering in november en een eendaagse 4all \TeX -cursus voor een 25-tal docenten in december.

Binnen het bestuur werd regelmatig en uitgebreid gediscussieerd over het nut van \TeX en NTG in het bijzonder, binnen de huidige en toekomstige wereld van documentverwerking. De discussie zou in 1997 worden voortgezet.

12 Diversen

- Van de TEX-NL discussielijst werd in 1996 wederom veel gebruik gemaakt. Zowel het aantal abonnees van deze listserver als ook het e-mail verkeer bleef zeer stabiel:

Eind:	1992	1993	1994	1995	1996
Aantal abonnees	162	180	175	181	183
Aantal e-mails	1097	1047	1242	1216	1105

- Het NTG-FGBBS Bulletin Board van Frans Goddijn werd door zowel NTG-ers als niet NTG-ers nog steeds regelmatig bezocht. Het FGBBS bevat naast de 4all \TeX CD-ROM, een indrukwekkende hoeveelheid aan (\LaTeX) \TeX materiaal.
- Vele leden maakten via ftp dankbaar gebruik van het CTAN mirror archief van Piet van Oostrum voor het verkrijgen van diverse soorten \TeX -materiaal. Daarnaast werd ook gebruik gemaakt van enkele andere ftp-server systemen in Nederland.

T_EX Users Group, gisteren, vandaag en morgen

Michel Goossens, TUG president

Het is weldra twee jaar geleden dat ik tot TUG president werd verkozen. Samen met mijn collega's van het TUG bestuur heb ik getracht gedurende die tijd TUG een metamorphose te laten ondergaan om in het bijzonder deze Organisatie aan te passen aan het Internettijdperk.

TUG's ledenaantal is in een vijftal jaren van 4000 tot zo'n 1500 gedaald (zie *TUGboat* 17,4 van eind vorig jaar, pagina 406 voor meer details) en daar kunnen we niet enkel de niet al te rooskleurige economische toestand laten voor opdraaien.

Wanneer we de evolutie van TUG en T_EX historisch willen bekijken, dan doen we dat best in de context van de parallelle evolutie in de computer technologie. We kunnen stellen dat tot in het midden van de jaren tachtig T_EX vooral gebruikt werd door wetenschappers op grote computersystemen (mainframes). De prijs van enkele honderden dollars voor een magneetband met T_EX software, die toen enkel te verkrijgen was bij TUG in Providence in de Verenigde Staten van Amerika (waar TUG zijn hoofdkwartier had tot eind 1992 en waar er zeven mensen—sommigen deeltijds—werkten !) was totaal te verwaarlozen vergeleken met de kostprijs van de computer zelf—een IBM, VAX, CDC of Cray—, die toen nog enkele miljoenen dollars waard was. Van globale netwerken was er nog geen sprake, en van WYSIWYG (*What You See Is What You Get*) tekstverwerkers nog veel minder.

Maar toen kwam de PC, en opeens werd het mogelijk een hele boel taken op zijn eigen persoonlijke werkpost (Mac, IBM PC, Atari, maar ook Sun, HP, enz.) te behandelen, en bovendien kon men met deze machines omgaan op een gebruikersvriendelijke (d.w.z. intuïtief visuele) manier. Professoren, studenten, secretaresses konden voor het eerst zelf hun teksten relatief pijnloos invoeren en op een gemakkelijke manier de constructie van de bladspiegel beheren. Het waren gouden jaren voor T_EX (en *nroff*) en inderdaad tussen 1987 en 1990 verdubbelde TUG's ledenaantal. Terzelfdertijd werden locale T_EX gebruikersgroepen opgericht, ondermeer in Nederland (NTG), Frankrijk (GUTenberg), Duitsland (DANTE), Groot Britannië (UKTUG). Deze groepen richtten zich vooral op het optimaal behandelen van typographische and linguïstieke specificiteiten eigen aan de respectievelijke talen en cultuurgebieden.

Omstreeks het begin van de jaren negentig is het globale Internetnetwerk ernstig komen opzetten, en het werd alsmear gemakkelijker (en voordeliger om tijdverlies te vermijden) om eerst kleine, en daarna ook grotere be-

standen via *ftp* (*file transfer protocol*) of het Web (afkorting van *World Wide Web*) van een Internetserver naar zijn eigen computer over te brengen. Zo staan, bijvoorbeeld, sinds 1993 de CTAN (*Comprehensive T_EX Archive Network*) servers kosteloos ter beschikking van de T_EX gemeenschap overal ter wereld, en vanaf 1995 werd het mogelijk via een gebruikersvriendelijke Web interface bestanden of documentatie van deze T_EX Web servers te bekomen. In 1993 kwamen ook de eerste T_EX-specifieke CDROM's op de markt (de pioniersrol van NTG met *4allT_EX* mag hier gerust onderstreept worden !), zodat de huidige T_EX-gebruiker met een Pentium PC (of eender welk ander juweeltje uit de multi-media hardware catalogus) heel gewoon alles wat hij/zij nodig heeft op een CD-tje of met een Webbrowser zonder moeite kan vinden. Het is bijgevolg echt niet meer nodig lid te zijn van TUG, alles ligt zo voor 't grijpen op de Internet of op een CDROM, die slechts enkele gulden kost. Ten hoogste is het nuttig lid te worden van een lokale T_EX gebruikersgroep, zoals NTG, om steeds op de hoogte te blijven van de laatste ontwikkelingen voor T_EX en, in ons geval, het Nederlands. Bovendien zijn deze „locale” groepen erg actief op het gebied van het aanpassen van T_EX aan locale behoeften en gebruiken—zoals het ontwikkelen van breekpatronen—, met het organiseren van seminaries en studiedagen in de moedertaal van de deelnemers, het vertalen van brochures en handleidingen, enz.

De basisgegevens voor „TUG toen” en „TUG vandaag” zijn dus helemaal anders en we moeten toegeven dat ook de „concurrentie”, d.w.z. de WYSIWYG tekstverwerkers, er heel wat beter op geworden zijn. De jongere generatie, die opgroeide met PC en MS Word of WordPerfect (mijn beide zonen van 17 en 19 jaar, bevoorbeeld) zweren enkel bij deze programma's en maken er inderdaad tamelijk aantrekkelijke meesterwerkjes mee. Men kan in een handomdraai simpele wiskundige formules of tabellen op het scherm (en het papier) toveren, en enkel voor ingewikkelde wiskundige uitdrukkingen is T_EX nog een *must*. T_EX's archaische, batch-georiënteerde, niet-intuïtieve syntax vergt een té grote intellectuele investering in vergelijking met de optimaal geïntegreerde „Office” programmasuites van Corel, Lotus, Microsoft, enz. Deze programma's maken het de gebruiker inderdaad mogelijk met één of twee muisklikjes een willekeurig deel van een venster op het computerscherm te selecteren en naar een ander venster over te brengen om het in een tekst op te nemen, een berekening te maken, een histogram te tekenen of een elektronisch bericht samen te stellen.

En hier zit het echte gevaar : het schijnt de laatste jaren erg moeilijk te zijn om „nieuw bloed” voor onze T_EX gemeenschap te vinden ; op de meeste T_EX bijeenkomsten zie je veelal oude kennissen, vertrouwde gezichten die je vijf of meer jaar geleden al ontmoette in dezelfde context. Maar de jeugd is meestal erg minimaal vertegenwoordigd, en als dan eens een twintiger opduikt om op het „Net” een vraagje te stellen, dan gaat het nog al te vaak om een student(e) wiens *supervisor* hem of haar „verplichtte” T_EX te gebruiken¹.

Wat kunnen we doen in deze situatie ? Het is wellicht zó dat HTML en de Web heel wat aantrekkelijker zijn om zich mee bezig te houden, i.h.b. indien men vlug resultaten wenst. Typographisch aantrekkelijke teksten zijn meestal de uitzondering dezer dagen. Het sleutelwoord is *informatiestroom* en het belangrijkste doel is de aandacht trekken van een maximaal aantal „gebruikers” (lezers) om „het bericht” door te geven. Hoe meer lezers, hoe groter de beloning. *McDonald* weggooi Webpagina’s van de Internet, met hun junk inhoud getekend in de schreeuwendste kleuren van de regenboog zijn overal legio. Verstandig en keurig opgestelde teksten, waar de vormgeving van de pagina de inhoud ondersteunt en het essentiële voortvloeit uit de eenvoud van de constructie zijn bijna onvindbaar. Overvloed in kleur, klank en vorm deugt niet. Het zijn eenvoud en harmonie die onze zintuigen op het essentiële concentreren. Al te veel Web-artiesten schijnen deze elementaire regels volledig te negeren.

En TUG in dit hele verhaal? Wel, TUG zal zich, zoals iedereen, moeten aanpassen. We zullen allen moeten leren de elektronische communicatiemiddelen optimaal te gebruiken—zonder in de flagrante anti-cultuur te vervallen die ik hierboven beschreef—om onze doelstellingen zo goed mogelijk te bereiken. Een korte lijst van initiatieven waar TUG plant mee verder te gaan in de onmiddellijke toekomst is : het regelmatig produceren van een nieuwe versie van de *TeXlive* CDROM, die het T_EX gebruikers gemakkelijk moet maken verschillende honderden megabytes met referentieprogramma’s en informatie gemakkelijk en continu bij de hand te hebben ; CTAN up-to-date houden en er indien nodig informatie aan toevoegen ; een referentie Web server opbouwen, waar men niet enkel terecht kan voor alles wat betreft T_EX en *Co*, maar ook voor tekstverwerking en informatiebeheer in het algemeen : PDF (Adobe’s *Portable Document Format*), SGML (*Standard Generalized Markup Language*), HTML (*Hypertext Markup Language*), XML (*Extensible Markup*

Language), PostScript, DSSSL (*Document Style Semantics and Specification Language*), Unicode, fonts, typografie, enz. In één woord TUG (en T_EX) moeten uit hun communicatieghetto treden en trachten zo vlug mogelijk de nieuwe technologieën te assimileren.

Een initiatief waar een steeds grotere groep binnen TUG over nadenkt is om *TUGboat* onafhankelijk van TUG te maken en het om te vormen in een echt internationaal tijdschrift over T_EX en documentbeheer. Men zou zich dan erop kunnen abonneren zonder lid te worden van TUG en we zouden dan ook artikels kunnen publiceren samen met andere (SGML, typografische, HTML, ...) gebruikersgroepen. Ik denk dat *TUGboat*’s autonomie een nodige stap is om te verzekeren dat deze publicatie hét referentieblad van de T_EXgemeenschap blijft, wat we ook met TUG zou kunnen gebeuren. Want T_EX heeft nog een mooie toekomst, met een hele reeks uitstekende en verreikende initiatieven, zoals het *Omega* project voor een verstandige integratie van Unicode en typografie, pdf2T_EX en dvipdf voor een interface met PDF en de wereld van de hypertext, en natuurlijk ook eT_EX, NTS en L^AT_EX3, die althans een meer klassiek pad bewandelen.

Ik wil dus mijn laatste woorden die ik tot jullie richt als president van TUG op een positieve noot eindigen. Vooreerst hou ik eraan NTG te danken voor hun belangrijke bijdragen voor T_EX. Er waren niet enkel het 4allT_EX CDROM pioniersproject en de uiterst geslaagde EuroT_EX’95 bijeenkomst in Arnhem in september 1995. Er is ook de continue inzet van al de NTG leden, die individueel of collectief steeds klaar staan om de wereldwijde T_EXgemeenschap bij te staan. In denk in het bijzonder aan de steun van NTG aan het *TeXlive* CDROM initiatief, het jaarlijks TUG *bursary fund*, de vele artikels voor *TUGboat* en de tientallen T_EX macros en programma’s die leden van NTG hebben ontwikkeld : té veel (en te gevaarlijk) om ze allemaal expliciet te vermelden—ik zou iemand ongewild kunnen vergeten. De uitstekende contacten met het NTG bestuur en het feit dat we immer konden rekenen op de aanwezigheid van een of meer NTG bestuurleden op onze TUG bijeenkomsten was iets wat we uiterst geapprecieerden. Ik ben ervan overtuigd dat TUG en de internationale T_EX gemeenschap ook in de toekomst op NTG zullen kunnen rekenen. Ik wens NTG van allerharte een gouden toekomst en hoop enkelen van jullie persoonlijk de hand te kunnen drukken op TUG’97 in San Francisco begin augustus of op EuroT_EX’98 eind maart volgend jaar in Saint Malo (Frankrijk).

¹Om te trachten meer jongeren voor T_EX te interesseren heeft GUTenberg het initiatief genomen om studenten 50% korting te geven op alle GUT bijeenkomsten en ze zullen trachten dat ook te doen voor EuroT_EX’98 volgend jaar.

Bericht van FGBBS — Voorjaar 1997

Frans Goddijn
fg@fgbbs.iaf.nl

Samenvatting

Hoewel “de loop eruit is” in de BBS-winkelstraten, houdt FGBBS stand als het rustige knopenwinkeltje dat met enige regelmaat wordt bezocht door een kleine maar trouwe klantenkring uit wijde omgeving.

1 Afstoffen

Nog steeds komt er bijna elke dag wel iemand op FGBBS kijken. Met een gemiddelde van 35 bellers in de maand zie ik soms volstrekt onbekende namen inloggen. De ene keer kijkt iemand alleen wat rond, haalt een niet-specifieke \TeX file op, en dan komt iemand enkele dagen achtereen langdurig “winkelen”, en worden er heel gericht enkele grote files, of een dozijn kleinere, uitgezocht en meegenomen.

De voortdurende bereikbaarheid (FGBBS is in principe altijd geopend) en de mogelijkheid voor iedereen om zonder drempel onbeperkt te kunnen downloaden is een sterke kant van het systeem. Bovendien, nu veel Internet providers te kampen hebben met trage verbindingen, is een modemverbinding met FGBBS vaak beduidend sneller voor het “overpiepen” van grote files.

Andere voordelen, zoals de overzichtelijkheid en actualiteit in vergelijking met de CD-ROM als distributiemedium voor software en informatie, zouden meer moeten worden uitgebaat. Ten dele ben ik daarvoor afhankelijk van anderen (het $4\TeX$ -team zou bijvoorbeeld af en toe een update naar FGBBS kunnen sturen), maar ik ben ook zelf meer met afstoffen en oppoetsen bezig dan voorheen. Als abonnee op de CTAN-ANN lijst zie ik aankondigingen van nieuwe files op het CTAN archief voorbijkomen, en een aantal daarvan heb ik de laatste maanden opgehaald en via FGBBS beschikbaar gesteld.

Voorts dient het systeem voor het in besloten kring verspreiden van mailinglijsten, zoals die ten behoeve van de bestelling van de $4\TeX$ -CD. Via deze lijst en ook door middel van een (password beveiligde) optie via interactief inbellen op FGBBS kan Gerard van Nes controleren wie er een CD hebben besteld.

Op het gebied van publiciteit kan voor FGBBS meer worden gedaan. Het was duidelijk merkbaar dat er meer bellers waren nadat het projekt van de nieuwe Nederlandse spellingswoordenlijst in de pers en op Internet onder de aandacht was gebracht en ook reageren mensen vanuit FIDOnet als er op de FIDO-lijst genaamd $\text{TEX}.028$ aankondigingen verschijnen van nieuwe files die van CTAN werden gehaald.

Enkele weken geleden is mede dankzij NTG-subsidie het systeem uitgebreid met enkele Gigabytes aan diskruimte en er kwam een sneller moederbord (P166) in een nieuwe (geruisarme) behuizing. Dank daarvoor!

2 Nieuwe files afgelopen 6 maanden

Algemeen

filenaam	bytes	toelichting
FGBBS.ARJ	56137	allfiles op FGBBS public hard disk (niet de CD's), in ARJ formaat
FGBBS.LST	181536	allfiles op FGBBS public hard disk (niet de CD's) TEXT formaat
FGBBS.TOT	2698	overzicht van totaal aantal files op FGBBS public HD
FGBBS.NEW	6277	nieuwe files op FGBBS public hard disk sinds 30 dagen
OS2TEX.ZIP	3945590	version 30 november 1996, a distribution of Donald E. Knuth's typesetting system TeX for the IBM operating system OS/2 Warp

DOC files, \LaTeX / \TeX manuals & tips

filenaam	bytes	toelichting
EPSLATEX.ZIP	123917	PostScript file, handleiding voor gebruik EPS graphics in \LaTeX

TeX, LaTeX, emTeX standard files

filenaam	bytes	toelichting
STARTEX.ZIP	869886	StarTeX package, a TeX format designed to help students write short reports and essays. It provides the user with a suitable set of comma

WORDS

filenaam	bytes	toelichting
AMS_NL5B.ZIP	480171	amspell woordenlijst januari 1997
WOORDMAX.ZIP	766488	woordenlijst januari 1997, groot formaat, IBM PC versie (met afbreekpunten)
WOOR-DEN.MAX	3364162	idem, ongezipt
WOORDMED.ZIP	516403	woordenlijst januari 1997, middel formaat, IBM PC versie
WOORDEN.MED	2131404	idem, ongezipt
WOORDMIN.ZIP	416262	woordenlijst januari 1997, klein formaat, IBM PC versie
WOORDEN.MIN	1623788	idem, ongezipt
WOORDEN.TXT	2551	tekst van Piet Tutelaers over het woordenlijst-project
README	944	Toelichting, Engels
WOORD.HTM	2193	HTML file over de woordenlijsten
AFLEID.HTM	1950	HTML file over afleidingen
HYPHENAT.HTM	1031	HTML file over afbrekingen
REGELS.HTM	2106	HTML file over de regels
SAMENST.HTM	777	HTML file over samenstellingen
VREEMD.HTM	874	HTML file over vreemde woorden
GBRAW.ARJ	793356	"Ruwe" lijst woorden van het Groene Boekje 1954
HARDEMAN.TXT	522	Toelichting door Gert Hardeman
WPLEX397.ZIP	1446961	Gert Hardeman's WP woordenlijsten, versie 5.0 / 5.1 / 5.2 / 6.0 WIN

Educational TeX/LaTeX files

filenaam	bytes	toelichting
VOORKPS.ARJ	28994	PS formaat: voorkaft van cursusboekje
DAG1-4PS.ARJ	238017	PS formaat: cursusedagen 1-2-3-4
LATEXDOC.HTM	13802	HTML versie van artikel "What is LaTeX all about?"
NLTEX96.HTM	5432	HTML versie van stukje uit MAPS over de cursus
COURSE.ZIP	555124	Inhoud van al het fraais uit de NTG-cursus door Bogusław Jackowski

Graphics package

filenaam	bytes	toelichting
OOREADME.TXT	4039	This File

CHANGES.TXT	21158	Log of changes to the packages.
GRAPHICS.INS	3602	Install file for docstrip.
COLOR.DTX	28201	Source for color package
GRAPHICS.DTX	56060	Source for graphics package
TRIG.DTX	9528	Source for trig package (required by graphics)
GRAPHICX.DTX	29423	Source for graphicx package (extension of graphics)
EPSFIG.DTX	4169	Source for epsfig package (extension of graphicx)
KEYVAL.DTX	10389	Source for keyval package (required by both the above)
DRIVERS.DTX	48841	Source for driver files for supported drivers.
TEXTURES.DEF	14428	Textures file supplied by Arthur Ogawa.
GRFGUIDE.TEX	35601	User Guide to all the packages in this bundle. WARNING: This file calls color and graphics packages without a driver option. You <i>*must*</i> s

LATEX2e Compatible styles

filenaam	bytes	toelichting
SUPERTAB.ZIP	14576	Braams & Jurriens Supertabular stylefile

Style Files collectie

filenaam	bytes	toelichting
ADVDATE.STY	5687	Manipulate the outcome of "today."
RECIPE.ZIP	58232	L ^A T _E X _{2ϵ} class file to typeset recipes
ENVLAB.ARJ	70460	print envelopes and labels (clever and complicated)
JKNAPPEN.ZIP	13108	Jörg Knappen's pakketten, voornamelijk op font-gebied, ook smartmn.sty voor slimmer gebruik van de "-"
PMGRAPH.ZIP	19977	A set of extensions to LaTeX picture environment, including a wider range of vectors, and a lot more box frame styles
COMMA.STY	2302	enables the printing of counter values with a separator every three digits, "12,345" or "12 345".
MINITOC.ZIP	42452	version #25, updates the support of norwegian
SHADOW.STY	4590	Mauro Orlandini writes: new, improved, version of my style file shadow.sty

WG: Educatie—een overzicht en oproep

Kees van der Laan

Samenvatting

Een overzicht van de activiteiten binnen en buiten de WG wordt gegeven, met daarnaast een suggestie.

Keywords: Educatie, WG

Op de laatste NTG bijeenkomst zijn de WG-en tegen het licht gehouden. Educatie had Jacko's cursus omhanden, dus was actief. In dit schrijven wil ik de educatie activiteiten tot nu toe op een rijtje zetten. Daarnaast wil ik aangeven wat er er mijns inziens centraal georganiseerd c.q. gestimuleerd zou moeten worden. Input, en ideeën zijn welkom.

In het verleden bleek een echte WG niet levensvatbaar in die zin dat er weinig bereidheid was tot investering in de overhead, c.q. werk te verrichten m.b.t. inventarisaties opzetten dan wel bijhouden. Men ging liever concreter lokaal aan de gang. Daar was en is niets op tegen.

Terugblik

In den beginne hadden wij tijdens de eerste NTG dagen de nostalgische cursussen. Daarna die van Victor en Amy. Vervolgens David's cursus—met als spin-off de MAPS special en dat David's boek misschien wat sneller uit is gekomen—en recentelijk Jacko met zijn METAFONT/PostScript. Ook is er aandacht besteed aan de logistiek via de 4AllT_EX CD-ROM cursussen. De speciale doelgroep van secretariaatmedewerkers is ook bediend. Diverse NTG leden hebben lokaal cursussen verzorgd al dan niet open voor gasten.

Voor (meta)font bleek geen markt. Recentelijk meldde Barbara Beeton zelfs dat METAFONT as such passé is en voortleeft in MetaPost.¹

Recentelijk worden in het reguliere (universitaire) onderwijs in Twente en Groningen studiepunten toegekend aan het zich bekwamen met L_AT_EX in een of ander verband.

Allemaal decentraal. Het enige centrale was het ervan weten dat het gebeurde.

Ooit heb ik een opzet geschreven om tot standardisatie van module te komen. Daarmee zou een leerpad mogelijk moeten worden via n'importe quoi welke TUG, EuroT_EX, dan wel LUG bijeenkomst. Dit is geen werkelijkheid

geworden, noch is de discussie daarover van de grond gekomen. Een reden—een van de ongetwijfeld velen—is dat docenten zelf willen bepalen wat zij doceren, gerechtvaardigd door de moeilijkheid realistisch vast te stellen wat er in een cursus aan de orde moet komen. Zelfs het overzicht van Bart Childs uit het verleden blijkt bij een terugblik niet adequaat, IMHO, met alle respect.

Het bijhouden van wie welke cursussen zou kunnen geven is achterhaald door de 'on demand' mogelijkheid van het net: als een cursus nodig is wordt deze als wens op het net gezet en wordt simpelweg hier-en-nu een docent gevraagd. Niks geen overhead. Gewoon vraag en aanbod.

Het gebied van (gestandaardiseerd) cursusmateriaal ligt ook al gevoelig. Docenten geven liever hun notities als boek uit. (Victor, David—althoewel zijn notes als MAPS special beschikbaar zijn gekomen—Clark, ...) Bovendien blijken leerdoelen niet vast te staan, en afhankelijk van de context. In het Nederlands had de L_AT_EX syllabus, uitgegeven bij het CWI, een functie. Wel is het zo dat cursussen door TUG/LUGs georganiseerd stimulerend werken voor de docent om zijn materiaal te verfraaien en ermee naar buiten te komen. De T_EXniques series van TUG hebben een educationele functie, evenals de MAPS specials of de FAQs zoals die van de UKTuG.

En tenslotte moeten wij niet vergeten dat de vraag-en-antwoord functie van het net een fantastische leerfunctie heeft. Wederom, niets geen overhead. Het naakte probleem en een hier-en-nu, to the point antwoord aub. As simpel as that.

Vooruitblik

Slaapt de WG? Nou nee, als er al sprake is van een WG.² Ik denk dat de WG infra-structureel moet werken en de vaardigheidsaspecten aan het marktmechanisme op het net moet overlaten.

Een cursus PostScript/PDF is nu relevant en haalbaar, IMHO, met alle respect, en de next step.

¹Ikzelf gebruik METAFONT als UI voor PostScript af en toe, weliswaar voor grafische doeleinden.

²Voorgestelde bijeenkomsten kwamen niet van de grond, vanwege andere prioriteiten en twijfel over de impact van de WG.

De nieuwe NTG WWW-pagina's

Erik Frambach & Hans Hagen

Samenvatting

Sinds kort zijn de WWW-pagina's van de NTG (<http://www.ntg.nl/>) geheel vernieuwd. Dit artikel geeft een overzicht van de ideeën achter de nieuwe opzet en uitvoering.

Keywords: WWW, NTG

1 Inleiding

Sinds Henk de Haan en Phons Bloemen in 1995 de NTG bij zijn eerste stapjes op het World Wide Web begeleiden is er veel gebeurd. Het WWW is volwassener geworden, de NTG is volwassener geworden, de inzichten zijn veranderd (verbeterd hoop ik), de mogelijkheden zijn groter geworden. Bovendien heeft de NTG sinds een tijdje zijn eigen Internet-domein ([ntg.nl](http://www.ntg.nl)) en draait daar een WWW-server en FTP-server die internationaal gezien mag worden.

Bij zo'n professionele opzet hoort natuurlijk een professionele inhoud en vormgeving. De oude WWW-pagina's voldeden echter niet meer aan die eisen. De WWW-pagina's zijn daarom grondig herzien.

De problemen met de oude pagina's kunnen als volgt samengevat worden:

- De opzet was onoverzichtelijk geworden;
- Er stond te veel verouderd materiaal;
- Er was geen consistentie qua inhoud, taalgebruik of vormgeving;
- Er ontbrak veel belangrijke informatie;
- Het geheel zag er niet erg aantrekkelijk uit.

2 Hoe het zou moeten

Om ideeën op te doen over wat goed en slecht is bij het opzetten van een set WWW-pagina's zijn we bij andere TeX-gebruikersgroepen gaan rondkijken.

De Duitsers, te vinden op <http://www.dante.de/>, heten de bezoeker hartelijk welkom op hun site. De home page straalt de spreekwoordelijke Duitse degelijkheid uit, is compleet en overzichtelijk. Al heel wat beter dan onze eigen site.

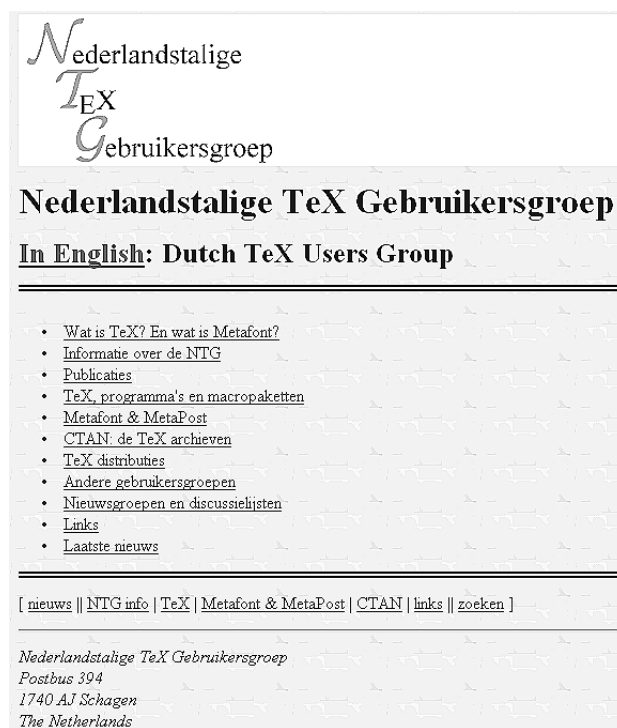
De Engelse site, <http://www.tex.ac.uk/UKTUG/>, besteedt daarentegen erg veel aandacht aan haar doelstellingen. Bezoekers met een klein scherm en een trage verbinding krijgen allereerst een prent te zien. Een rommelige titel, weinig informatie, nee zo moest het zeker niet.

De Franse site was niet helemaal up-to-date en straalde het bekende Franse chauvinisme uit: een minimum aan

verwijzingen naar elders. Je zou dat eerder van Hollanders verwachten, maar dit is waarschijnlijk de enige site waar men meteen bedragen noemt. Wie er eens wil kijken, <http://www.ens.fr/gut/> is het adres.

De binnenkomst bij TUG (<http://www.tug.org/>) is eigenlijk wat teleurstellend, zeker als we bedenken dat het daar ooit begon en dat het WEB daar is gesponnen. Als we moeten afgaan op wat daar wordt vermeld... Nee, men had beter kunnen volstaan met een verwijzing naar Dante.

En onze eigen site, wel, we hebben het al gezegd, alle genoemde slechte eigenschappen kwamen daar samen. We boden wel veel informatie maar de toegankelijkheid en de kwaliteit kon beter. En wat te denken van die teller. Zoals met wel met meer sites het geval is, was duidelijk te zien dat het steeds uitbreiden de kwaliteit niet had bevorderd.



Figuur 1: De nieuwe hoofdpagina.

3 De nieuwe opzet

Na deze inventarisatie hebben we voor onszelf de volgende opdracht geformuleerd:

- Maak een overzichtelijke structuur die steeds in beeld blijft;
- Verzin een attractieve vormgeving die je vervolgens overal toepast, maar hou het sober;
- Verwijder verouderd materiaal;
- Voeg ontbrekende informatie toe;
- Zorg ervoor dat de download-tijd en pagina-opbouwtijd kort blijft door:
 - zo weinig mogelijk plaatjes te gebruiken, en plaatje klein van omvang te houden (klein in de betekenis van aantal kilobytes! geanimeerde plaatjes kunnen beslist niet!);
 - niet te veel tekst per pagina te geven;
 - geen tabellen toe te passen;
 - geen 'frames' te gebruiken;
 - geen Java te gebruiken;
 - geen geluid te gebruiken;
 - geen tellers te gebruiken;
 - geen video te gebruiken;
- Zorg voor toegevoegde waarde:
 - maak alle MAPSen elektronisch beschikbaar;
 - publiceer de complete MAPS-bibliografie (zie ook het artikel 'bib2html' elders in deze MAPS);
 - maak alle MAPS-artikelen afzonderlijk beschikbaar in PDF;
 - maak het mogelijk te zoeken in de MAPS-bibliografie;
 - maak het mogelijk te zoeken op de hele NTG-site.

Herman Haverkort, Frans Goddijn

The Scenario -- in Three Versions; hmapwerk does it, English, MAPS 14, 1995, pp. 156-162

keywords: paragraph, mark, distribution, select

abstract: During work towards a flexible document as a continuous report on a wide variety of contacts for the Meridian Arts Ensemble in New York, Frans Goddijn felt the need to tag and mark certain paragraphs for specific groups of readers. Herman Haverkort wrote a package for LaTeX2_ε, hmapmark, which facilitates this by offering the possibility to set various signs next to paragraphs. This article presents hmapmark, gives examples of its use and a short manual. For the hackers among us some of the TeXtrical tricks involved behind the scenes are glanced at.

14-42 in PDF (129 KB)

Bernd Raichle

Sorting in TeX's Mouth, English, MAPS 14, 1995, pp. 163-168

keywords: !!

abstract: TeX's macro processor, the so-called mouth, can be used to perform very complex tasks. Because this part of TeX's programming language is as powerful as a Turing machine, it is possible to implement algorithms using only TeX's mouth. I will show how sorting algorithms can be implemented in a straight-forward and very elegant and understandable way using only macros and macro expansion TeXniques.

14-43 in PDF (81 KB)

Kees van der Laan

One by one the guests arrive, English, MAPS 14, 1995, pp. 169

keywords: education, macro writing, philosophy, plain TeX

abstract: A plea is made for writing macros in plain TeX, sufficiently documented to be used with all flavours of TeX.

14-44 in PDF (23 KB)

Figuur 2: Een stukje uit de MAPS-bibliografie.

4 Realisatie

Voor het realiseren van deze doelstellingen hebben we zo veel mogelijk gebruik gemaakt van de bestaande documenten. Waar nodig hebben we nieuwe stukken bijgeschreven.

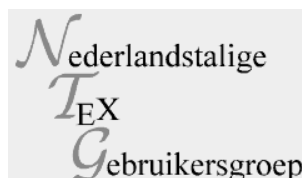
4.1 Schaduw-draaien

Op Eriks persoonlijke WWW-server heeft de hele set een tijdje proef gedraaid. Vervolgens is alles (circa 100 megabyte inclusief alle PDF-documenten) op cd-rom gezet en naar Piet van Oostrum, de beheerder van de NTG WWW-server, opgestuurd. Piet heeft vanaf de cd-rom alles op de definitieve plaats geïnstalleerd.

Voor het maken van de pagina's is een gewone ASCII-editor gebruikt, geen speciale HTML-editor of export-functie vanuit een tekstverwerker. Vandaar dat de HTML-code er zo keurig netjes uitziet (dit is geen grap) en voor een ieder als voorbeeld kan dienen voor eigen pagina's. Overigens is de HTML-code wel gecontroleerd met verschillende HTML-validatieprogramma's.

4.2 Het NTG-logo

Na het bekendmaken van de test-site stuurde Bernd Raichle mij een alarmerend bericht: hij ontwaarde in ons NTG-logo een verouderde \mathcal{T} ! Donald Knuth had al jaren geleden zijn Calligraphic font ge-update waarbij het schreefje van de T was verdwenen, dus moesten wij ook het nieuwe font gebruiken, zo redeneerde Bernd. Het NTG-bestuur heeft echter anders beslist. We beschouwen het NTG-logo als een kunstwerk dat destijds gemaakt is met de middelen die er toen waren. Dat er inmiddels dingen veranderd zijn doet niet terzake. We vragen immers Rembrandt ook niet om de Nachtwacht opnieuw te schilderen omdat er nu betere verfsoorten bestaan.



Figuur 3: Het NTG-logo.

4.3 Frivoliteiten

Om de pagina's wat attractiever te maken hebben we ons enkele frivoliteiten gepermitteerd:

- We hebben onze eigen kleuren ingesteld voor de achtergrond en voor hyperlinks. Het zijn subtiele blauwgroene tinten, min of meer aansluitend bij de kleuren van de MAPS-omslag. Met dank aan Taco Hoekwater.
- Taco leverde ook het 'TeXnische' achtergrondplaatje dat we gebruiken; uitvergroot ziet dat er zo uit (het contrast is hier verhoogd):



- We hebben lijntjes toegevoegd in de Nederlandse driekleur. Met dank aan Siep Kroonenberg, die ook het NTG-logo geschikt maakte voor WWW-presentatie.

5 Wat nog moet gebeuren

Enkele zaken zijn (op het moment dat we dit schrijven) nog niet verwezenlijkt:

- Zoeken in de NTG-pagina's;
- Zoeken in de MAPS-bibliografie;
- De MAPS-bibliografie is nog niet compleet. Op veel plaatsen ontbreekt nog *abstract* en/of *keywords*: stuur a.u.b. aanvullingen!
- Enkele pagina's zijn nog (vrijwel) blanco.

6 Conclusie

We hopen dat de NTG met de nieuwe opzet weer een tijdje vooruit kan. Ideeën, op- en aanmerkingen zijn natuurlijk van harte welkom. Onderschat uw rol als bezoeker niet! Wijs ons op fouten. Niets is zo hinderlijk als een verkeerde link en niets is zo frustrerend als achteraf horen dat iemand wist waar Abraham de mosterd haalde maar vergeten wat dit op zijn site te vermelden.

Gezeefd uit de TEX-NL discussielijst

Frans Goddijn
fg@fgbbs.iaf.nl

Samenvatting

In 1993 publiceerde Philippe Vanoverbeke in de MAPS tweemaal een selectie van berichten uit de TEX-NL lijst. Nu, een poosje later, wordt de draad weer opgepakt. Een aantal oplossingen, hints en gouden tips over onderwerpen waarvan je weleens denkt: “hoe zat dat ook alweer?”

Deze informatie is, samen met nog veel meer van dergelijke waardevolle berichten uit TEX-NL, ook na te lezen op FGBBS in het berichtengebied FGBBS.Archive.

Opvallend is dat in bijna alle gevallen het snelste en definitieve antwoord wordt gegeven door Piet van Oostrum (piet@cs.ruu.nl, <http://www.cs.ruu.nl/piet>), zonder wie TEX-NL niet het zeer nuttige en snelle medium zou zijn dat het nu is. OOSTRUM-NL zou ook een goeie naam zijn ☺

1 De linkerbovenhoek

Dr.ir. R.A. Hakvoort (RUDI@sepa.tudelft.nl) vroeg:

Ik ben op zoek naar een L^AT_EX (2.09) style-file om de linkerbovenhoek van een tabel als volgt vorm te geven:

<table style="border-collapse: collapse; width: 100%; border: none;"> <tr> <td style="border: none;"> \ item1 </td> <td style="border: none;"> </td> <td style="border: none;"> </td> <td style="border: none;"> </td> </tr> <tr> <td style="border: none;"> \ \ </td> <td style="border: none;"> </td> <td style="border: none;"> 1 </td> <td style="border: none;"> etc. </td> </tr> <tr> <td style="border: none;"> item2 \ </td> <td style="border: none;"> </td> <td style="border: none;"> </td> <td style="border: none;"> </td> </tr> <tr> <td style="border: none;"> -----</td> <td style="border: none;"> </td> <td style="border: none;"> </td> <td style="border: none;"> </td> </tr> <tr> <td style="border: none;"> A </td> <td style="border: none;"> </td> <td style="border: none;"> ... </td> <td style="border: none;"> </td> </tr> <tr> <td style="border: none;"> -----</td> <td style="border: none;"> </td> <td style="border: none;"> </td> <td style="border: none;"> </td> </tr> </table>	\ item1				\ \		1	etc.	item2 \				-----				A		...		-----			
\ item1																								
\ \		1	etc.																					
item2 \																								

A		...																						

En prompt kwam het antwoord van Piet van Oostrum (piet@CS.RUU.NL):

Je kunt een picture omgeving voor het eerste vakje gebruiken. Het nadeel is dat je de breedte moet schatten.

Als je een postscript printer hebt en pstricks dan kun je zo iets doen:

```
\usepackage{pst-node}
\begin{document}
\begin{tabular}{|c|c|c|}
\hline
\pnode{a}\hspace{5mm} item1 \\\
& 1 & 2 \\\
item2\hspace{5mm} \pnode{b}\ncline{-}{a}{b} \\\
\hline
A & A1 & A2
\end{tabular}
```

evt. kun je de \pnode nog een (x,y) argument geven om de eindpunten van de lijn beter te positioneren.

2 Floating Figures

Kay Behnke (behnke@mpi.nl) vroeg:

I have the following “problem”. I have an Encapsulated Postscript figure which I would like to include into a text. The figure does not fill the full width of the page so that I would like to set text beside the figure. Well, on page 150 of **The L^AT_EX Companion**, the package of T.J. Reid is described which would fit to my problem. But to get sure: Do you know of anything else? How would you handle such a problem?

Het antwoord kwam opnieuw van Piet van Oostrum:

There is a L^AT_EX_{2 ϵ} version called floatflt. I always use picins because I find it better.

There are a couple of style files that purport to do this. But they all have their limitations because the T_EX machine isn't really designed to solve this.

flow.sty, floatfig.sty (for L^AT_EX_{2 ϵ} : (floatflt.drx), figures.sty, picinpar.sty, picins.sty (from a package picins), window.sty & wrapfig.sty.

Here is a brief impression (from the source code, and some experiments):

2.1 figures.sty

```
\rightfig{width}{height}{caption indication}{caption}
    ... paragraph text ...
```

similar \leftfig

You must specify the width and height of your picture yourself. No possibility to include it in table of figures/tables without additional effort.

2.2 picinpar.sty

```
\begin{window}[#lines before, l|r|c, picture contents, caption]
    ... paragraph text ...
\end{window}
```

Gives the possibility to specify that the paragraph must have advanced a number of lines before the picture begins. Width and height will be deduced from the contents. The picture may span several paragraphs. The picture can be positioned left, right or in the center of the page.

Does not work with the amstex.sty option.

NOTE: I have studied the code of this package extensively and it is badly programmed. There is a lot of redundant code and worse, it is buggy. E.g. if you set

```
\box0=\hbox{1}\parskip=\ht0
```

and you have plain text in the paragraph around the figure you will loose the text!!!

Moreover if you center the picture and the lines at the left and right of the picture don't have the standard height (e.g. because you have a formula like A^n_i in one of the parts) they don't line up properly.

I also encountered a real-life example where it swapped a few lines.

2.3 wrapfig.sty

```
\begin{wrapfigure}[height of figure in lines]{l|r}{width}
    {figure, caption etc.}
\end{wrapfigure}
```

Height can be omitted and will be calculated in that case. Will be counted as figure.

2.4 flow.sty

```
\flow[L|R]{paragraph text}{figure box}
```

figure must be in a T_EX box. A frame will always be drawn. No captions.

2.5 floatfig.sty

```
\begin{floatingfigure}{figure}
\end{floatingfigure}
```

Will be counted as figure. For L^AT_EX_{2 ϵ} there is a new version floatflt.dtx that also does tables.

2.6 window.sty

```
\windowbox[toplines][inwindow: contents][ratio: 1 r]
... paragraph text ... \par
```

Gives the possibility to place a picture on an arbitrary location in a paragraph (e.g. on 2/3 of the width). No captions or table of figures inclusion. According to the author there are problems when inserting boxes at the end of a page or at the beginning of a section. This style is no longer supported.

2.7 picins.sty

picins.sty is part of a large package that allows inclusion of pictures (e.g. with shadow boxes, various MS-DOS formats etc).

```
\parpic(width,height)(x-off,y-off)[Options][Position]{Picture}
Paragraph text....
```

All parameters except the Picture are optional. The picture can be positioned left or right, boxed with a rectangle, oval, shadowbox, dashed box, a caption can be given which includes it in the List of figures.

This is the only package that I tried that correctly works inside an enumerate/itemize item. It does not, however work with enumerate/itemize besides the picture (i.e. started after the \parpic command) but neither does any of the other packages as far as I could discern.

The only disadvantage for some people will be that the documentation is in German. I have included a summary in English as picins.txt on CTAN.

Picins looks like the best and nicest to me. I would like to receive additions or corrections to the above list.

3 Cirkel om symbool

Harmen van den Berg (*H.vandenBerg@math.utwente.nl*) vroeg:

Hoe kan ik cirkels om symbolen heen zetten, zoals bij \copyright (©) of \circledS, maar dan voor willekeurige symbolen? (In mijn geval gaat het om de getallen 1, 2, 3, enz. in een cirkel.) Is er een font dat dit soort cirkels kent?

3.1 Eerste antwoord

ook weer van Piet...:

In L^AT_EX: ① (\textcircled{\footnotesize 1})

\textcircled is een ongedocumenteerd L^AT_EX “accent”.

cmsy heeft een cirkel die aan de grote kant is. lcircle heeft diverse cirkels. wasy heeft een cirkeltje dat even groot is als degene die in \oplus etc gebruikt wordt.

3.2 Tweede antwoord

door Dr. J.A. van de Griend, *vdgriend@WI.LEIDENUNIV.NL*:

In mijn aantekeningen over intervalaritmietiek maak ik gebruik van de macro \intcircle. Deze zet het symbool #1 in een cirkel (b.v. een ‘+’) en laat het resultaat fungeren als een binaire operator. Je kunt de cirkels natuurlijk ook wat groter maken.

```
\font\intvalz=cmsy10
\def\medcircle{\hbox{\intvalz\char'015}}
\def\intcircle#1{
  \def\loc{\raise1pt\hbox{\scriptscriptstyle{#1}$}}
  \setbox4=\hbox{\loc}
  \let\cir\medcircle
  \def\laag{0.03em}
  \setbox3=\hbox{\cir}
  \mathbin
  {\hbox{\kern0.5\wd3\kern-0.5\wd4\rlap{\loc}\kern-0.5\wd3
  \kern0.5\wd4\lower\laag\cir}}}
```

$\intcircle{3}$ geeft ③

3.3 Wat ook kan

Herman Haverkort schreef er niks over in TEX-NL maar hij *had het kunnen doen*: voor het flexibel omcirkelen van allerlei dingen heeft hij het pakket `hhflxbox` geschreven, dat op zijn beurt `hhunits.sty` en `hhqueue.sty` aanroept. Dit geeft de volgende optie. Je maakt een commando zoals bijvoorbeeld

```
\newcommand{\omcirkel}[1]{\ringbox omcirkel:#{#1}}
```

en vervolgens kun je met dit commando van alles omcirkelen, met als bijzonderheid dat de cirkel *na een tweede maal compileren* in alle gevallen even groot is:

en $\textcircled{1}$ en $\textcircled{123}$

4 Papier besparen

Robert van der Geest (R.A.B.vanderGeest@math.utwente.nl) vroeg:

Met het programma `dvidvi` is het mogelijk om twee A5-pagina's samen op een landscape A4-pagina te plaatsen. Ik zou nu graag twee A4-pagina's op een landscape A4-pagina plaatsen, d.w.z. verkleind naast elkaar. Mijn tactiek hiervoor is om met `dvidvi` twee A4-pagina's op een landscape A3-pagina te plaatsen en vervolgens deze A3-pagina verkleind uit te printen. Deze tactiek werkt echter niet! `dvidvi` slaagt er niet in om twee A4-pagina's op een A3-pagina te plaatsen.

Waarop heel snel het antwoord van **Mr. TEX-NL** verscheen:

```
From: Piet van Oostrum <piet@cs.ruu.nl>
Bij mij wel.
dvidvi -m 2:0,1'(21cm,0)' rep.dvi temp.dvi
dvips -0 -0.5in,-0.5in -x 707 -t landscape temp.dvi > temp.ps
```

Een ander, Johan Wevers (johanw@vulcan.xs4all.nl) ging er ook op in:

Het zou natuurlijk helpen als je aangaf wat voor printer je hiervoor meestal gebruikt. Als dat een hplaserjet compatible is kun je ook met `\mag=707` de oppervlakte verkleinen en dan met de emTeX drivers 2 pagina's landscape naast elkaar zetten. Ikzelf gebruik `\mag=500` en zet 4 pagina's op een vel als ik iets wil verkleinen. Dan moet je wel die verkleinde fonts aanmaken.

Je kunt de printerfile dan aanmaken met

```
dvihplj @lj_500.cnf @4pagelj.res /ptnul %1 %1.lj %2 %3 %4 %5 %6 %7 %8 %9
```

(of direct de printer aanroepen). `4pagelj.res` ziet er zo uit:

```
% 4page.res
/cp4
/ci1 /cx-0.6in /cy-1in
/ci2 /cx3.7in /cy-1in
/ci3 /cx-0.6in /cy4.5in
/ci4 /cx3.7in /cy4.5in
/ta=-0.6
```

waarin je de waarde van `/ta` aan moet passen aan jouw printer, anders kan de tekst te hoog of te laag komen.

5 Illegal footnotes?

Gilbert van den Dobbelsteen (GILBERT@login.iaf.nl) schreef:

Ik heb een probleempje, zie onderstaande constructie. Er verschijnt wel een '1' in superscript, maar geen voetnoot onderaan de pagina (of waar dan ook). Heb ik hier te maken met een illegal environment waarvoor ik als oplossing `HH styles` kan gebruiken?

```
\begin{description}
\item[Foutje]{footnote{Niet te zien deze voetnoot}} De voetnoot wordt
niet afgedrukt.
\end{description}
```

Het antwoord van Piet van Oostrum:

Voetnoten in boxen kunnen verdwijnen. Je moet dan de combinatie `\footnotemark` (in de box), en `\footnotetext` (buiten de box) gebruiken. Het item van een description wordt ook in een box gezet, vandaar.

```
\item[Foutje\footnotemark]\footnotetext{Wel te zien}
```

6 Woordafbreking uit

Rob Lahaye (LAHAYE@AMOLF.AMOLF.NL) schreef:

Is het mogelijk om in een deel van mijn tekst de hyphenation helemaal uit te zetten zodat daar geen woorden worden afgebroken?

Piet van Oostrum antwoordde:

Gebruik de TeX constructie: `\lefthyphenmin=1000`

Rob Laheye weer:

Als ik deze waarde aanpas, wil ik hem daarna ook weer terug zetten. Wat is de waarde van deze variabele normaal. Ik kan dat nergens vinden. Hoe kom ik daar achter? hierover staat niets in **The L^AT_EX Companion!**
Verder heb ik ook gemerkt, dat ik niet weet hoe ik een `length`-variable kan uitprinten, zodat ik erachter kom wat de waarde is. Wie weet hoe dat kan?

Piet van Oostrum antwoordde:

De normale waarde hangt van de taal af. In de meeste gevallen 2, soms 3. Je kunt evt. het hele stuk warin niet afgebroken moet worden in een group zetten, dan blijft de verandering lokaal. Een fraaie manier om dit te doen is:

```
\newenvironment{nohyphen}{\lefthyphenmin=1000}{}
```

Misschien wil je ook nog wel `\sloppy` of `\raggedright` erbij om overfull boxen e.d. te vermijden.

```
\begin{nohyphen} ...
\end{nohyphen}
```

Wat je ook kan doen is de oude waarde saven:

```
\newcount\oldlefthyphenmin
\oldlefthyphenmin=\lefthyphenmin
\lefthyphenmin=1000
```

en na het nietafbreekstuk:

```
\lefthyphenmin=\oldlefthyphenmin
```

Waar je in ieder geval op moet letten is dat er een alineaovergang zit voor elke wijziging.

Wat versta je onder uitprinten? Ik neem even `\parindent` als voorbeeld van een lengte:

Als je zegt

```
\showthe\parindent
```

dan komt het in je logfile (en als je interactief werkt moet je een return intikken),

Als je zegt

```
\the\parindent
```

dan komt het in je tekst te staan.

7 Wit boven & onder hoofdstuktitel

Jan Loos vroeg:

Is it possible to move the whole title including Chapter or Mychapter over the page, so that it leaves less white space on top?

Johannes Idsø antwoordde:

To change the white space after or before the the chapterheading do this:

1. Save the following as `nn.sty` in `c:\texfiles\texinput`

```

%%%%% nn.sty starts here %%%
\renewcommand{\chapter}{\@startsection
  {chapter}%                %{name}
  {0}%                      %{level}
  {\z0}%                    %{indent}
  {2.5ex plus -1ex minus -.2ex}% %{beforeskip}
  {.2ex plus .2ex}%        %{afterskip}
  {\Huge\bf}}%            %{fonts etc.}
%%%%% nn.sty ends here %%%

```

2. Put this in your preamble:

```
\usepackage{nn}
```

Now it is easy to alter the white space after the chapterheading. Try for instance:

```
{20ex plus .2ex}%      %{afterskip}
```

in the `nn.sty` file and see the difference. Control the white space on top of the page with the `beforeskip` parameter.

The same commands can be used on sections and subsections, but then you have to use the name and the level accordingly. (section is level 1).

8 Section titel links-rechts opschuiven

Niels C.C.M.Moes (C.C.M.Moes@IO.TUDELFT.NL) vroeg:

Ik gebruik $\text{\LaTeX}2_{\epsilon}$, en documentclass `report`. Ik wil in de linker marge de sectienummering plaatsen, waarbij het eerste (linker) character op een vaste horizontale positie staat. Dat lukt gemakkelijk met:

```

\renewcommand{\section}{\@startsection
{section}%
{1}{-2cm}% negatieve indent van 2cm
enz.}

```

De nummering is dan weliswaar, ten opzichte van de linkermarge, 2cm naar links verschoven, maar ik niet bereiken, dat de section-titel automatisch precies op de marge begint.
Kan iemand mij hiermee helpen?

Piet van Oostrum, kort en bondig:

```

\makeatletter
\renewcommand{\@secntformat}[1]{\hbox to2cm{\csname the#1\endcsname\hfill}}
\makeatother

```

De hele `\renewcommand{\section}{\@startsection etc.` kun je weglaten als je het volgende doet:

```

\makeatletter
\renewcommand{\@secntformat}[1]{\llap
  {\hbox to2cm{\csname the#1\endcsname\hfill}}}
\makeatother

```

of:

```

\renewcommand{\@secntformat}[1]
  {\hbox to 0pt{\hskip-2cm\csname the#1\endcsname\hfill}}

```

9 Een ander soort description environment

Kees de Bruin vroeg:

In een rapport dat ik aan het schrijven ben, moet ik een tabel maken met behulp van een `description` environment, maar alle tekst moet netjes onder elkaar staan. Ik weet dat het mogelijk moet zijn, maar omdat ik voor de komende dagen geen toegang heb tot de documentatie, kan ik dat niet zelf opzoeken. Het moet er dus als volgt uit zien:

```
\begin{newdescr}{WWWW}
\item[A] tekst...           A      tekst
\item[AB] tekst            AB      tekst
\item[ABC] tekst           ABC      tekst
\item[ABCD] tekst          ABCD     tekst
\end{newdescr}
```

Het argument moet aangeven hoeveel moet worden ingesprongen om de tekst netjes uit te lijnen. Ook bij meerdere regels moet de tekst netjes op dezelfde positie beginnen.

Piet antwoordde:

```
\newcommand{\mylabel}[1]{\mbox{#1}\hfil}
\newenvironment{newdescr}[1]
{\begin{list}{}%
{\renewcommand{\makelabel}{\mylabel}%
\settowidth{\labelwidth}{#1}%
\setlength{\labelsep}{0pt}% aangenomen dat #1 breed genoeg is
\setlength{\leftmargin}{\labelwidth}}%
{\end{list}}}
```

```
A      tekst..
AB      tekst
ABC     tekst
ABCD    tekst
```

10 Regelafstand

Maarten Gelderman (mgelderman@ECON.VU.NL) vroeg:

Is het mogelijk om de regelafstand direct in punten aan L^AT_EX door te geven, of kan het alleen via `\baselinestretch` ? `\baselineskip` werkt niet.

Piets antwoord:

`\baselineskip` kan in principe ook, maar dit wordt op regelmatige tijden door L^AT_EX teruggezet (in bij ieder geval bij iedere fontwisseling, en bij diverse initialisaties).

De standaard manier in L^AT_EX om de `\baselineskip` te zetten is met het `\fontsize` commando; dit heeft de vereiste `\baselineskip` als tweede parameter. Dus bijv.

```
\fontsize{11}{13pt}\selectfont
geeft een \baselineskip van 13pt.
```

Als je dit voor een huisstijl wilt doen dan is een class, option of style file de aangewezen plek. Dan zie je vaak commando's van de vorm:

```
\newcommand\large{@setfontsize\large@xiipt{14}}
14 is dan de \baselineskip (in pt).
```

11 Hakken met pijlen

Erik Van Eynde (Erik.VanEynde@cc.kuleuven.ac.be) zocht naar een bepaalde soort pijlen. Piet Tutelaers, Taco Hoekwater en Johan Wevers kwamen met diverse oplossingen. Erik vat ze samen:

Sorry, Johan en Taco hadden me verkeerd begrepen. Ik wens een links pijl samen te combineren met een pijl naar boven. Maakt uiteraard weinig uit of het linkse, rechtse pijl naar boven of onder wordt gebruikt. Het gaat om combinaties van karakters. De oplossing van Piet is onvoldoende (opwaartse pijl te laag en te veel naar rechts...)

Taco's oplossing is eenvoudig, maar resultaat is niet perfect (opwaartse pijl zou meer naar rechts moeten schuiven)...

`\rlap` is dat een L^AT_EX commando? De aangepaste oplossing van Johan blijkt het beste resultaat te geven. Amstex package is niet nodig (Taco).

Hieronder volgt een samenvatting van de verschillende mogelijkheden:

```
% Hakkerij met pijlen - Piet T.
\font\symbol=cmsy8
\newcommand{\leftuparrow}{\hbox{\symbol\char'040\kern-10pt\raise-.5ex%
  \hbox{\char'042}}}}

\def\leftuparrowa{\mathrel{\rlap{\$ \leftarrow}\uparrow}} % Taco H.

\def\lrrarrow{\sim\lower.2ex\hbox{\$ \rightarrow}\kern-2.4ex\raise.7ex
  \hbox{\$ \leftarrow}\sim}

\def\rlrarrow{\sim\lower.2ex\hbox{\$ \leftarrow}\kern-2.3ex\raise.7ex
  \hbox{\$ \rightarrow}\sim}

\def\luarrow{\sim\hbox{\$ \leftarrow}\kern-1.5ex\raise.3ex\hbox{\$ \uparrow}\sim}

\def\ruarrow{\sim\hbox{\$ \rightarrow}\kern-2.0ex\raise.3ex\hbox{\$ \uparrow}\sim}

$ \leftuparrow$           ↖           (Piet T.)
$ \leftrightharpoonup$   ↔           (Taco H.)
$ \leftuparrowa$         ↗           (Taco H.)
$ \lrrarrow$             ⇌           (Johan W.)
$ \rlrarrow$             ⇌           (Johan W.)
$ \ruarrow \quad \luarrow$ ↗ ↖       (Erik V.)
```

12 Superscript feature

P. Wackers (P. Wackers@LET.KUN.NL) vroeg:

Waarschijnlijk een simpele vraag, maar ik kan nog niet goed overweg met het nieuwe fontselectiemechanisme. Ik gebruik voor een tekst de pandorafonts (via het gelijknamige package). In de lopende tekst moeten soms letters of cijfers superscript afgedrukt worden. Dat is op zichzelf heel simpel. Ik doe:
 tekst tekst^{cijfer}
 en als ik het cijfer in romein wil, zet ik er `\rm` voor. Maar doordat L^AT_EX hier mathfonts aanroept, wijken de superscripte tekens in vorm af van die in de eigenlijke tekst. Ik zou dat graag voorkomen. Hoe herdefinieer ik het betreffende mathfont? (Ik had hetzelfde probleem met voetnootnummers, maar dat is dankzij Piet van Oostrum opgelost). Of is het in dit soort gevallen slimmer om een `\raisebox` te gebruiken?

Piet antwoordde: Pandora heeft geen echte math fonts, je moet dus zoveel mogelijk tekens “lenen” uit de tekst fonts.

Zo is het ook gedaan met Times, als je geen geld hebt om de TimesMath fonts in het mathptm package te kopen.

Mijn suggestie: maak een kopie van mathptm.sty en vervang de postscript fonts daarin door pandora.

Siep Kroonenberg (N. S. Kroonenberg@eco.rug.nl) kwam met deze:

L^AT_EX_{2_ε} heeft ook een `\textsuperscript` macro.

13 ZapfDingbats in any size

Dave Wapstra (davew@pi.net) schreef:

Ik zoek een eenvoudige manier om de ZapfDingbats in een bepaalde grootte te krijgen. Nu doe ik het zoals het hieronder staat, maar kan het misschien nog eenvoudiger?

```
\documentclass{article}
\usepackage{pifont}
\begin{document}
\fontfamily{pzd}\fontsize{50}{50}\fontencoding{U}\selectfont
\ding{238}
\end{document}
```

Piets antwoord:

```
{\fontsize{50}{50pt}\ding{238}}
```

(De tweede parameter van `\fontsize` hoort een lengte te zijn — wordt als `baselineskip` gebruikt, maar bij zo'n enkel symbool is dat natuurlijk irrelevant.)

Of je kan helemaal low-level gaan:

```
\font\zd=pzdr at 50pt \zd\symbol{238}
```

14 dvips -y vergroot geen plaatjes??

Gerard van Nes (*vannes@ECN.NL*) schreef:

Een collega maakt zijn dissertatie m.b.v. L^AT_EX (logisch). In bevroren versie 2.09 (gelukkig ben ik niet de enige...).

Plaatjes worden gemaakt via `psfig`.

Dissertatie moet vergroot worden aangeleverd (10/7) om later met 70% gereduceerd te worden.

Met `dvips -y` parameter wordt alles vergroot. Tekst gaat prima, echter de plaatjes blijven dezelfde maat houden.

Welke 'truc' kan ik het beste gebruiken om de plaatjes gelijk met de tekst mee te vergroten? Mogelijk zie ik wat over het hoofd.

Piet van Oostrum antwoordde:

Niet `psfig` gebruiken, maar `epsf` of `epsfig` (de laatste maakt het mogelijk om toch de `psfig` commando's te blijven gebruiken).

De *specials* die `psfig` gebruikt zijn niet bestand tegen schalen. Dat is ook de reden dat ik zo tegen het gebruik van `psfig` ben.

De *specials* die `epsf` gebruikt zijn dat wel. `epsfig` is een uitbreiding van `epsf` die de `epsf` *specials* gebruikt, maar met de user interface van zowel `epsf` als `psfig`. Je moet dan natuurlijk wel de L^AT_EX 2.09 versie van `epsfig` gebruiken (en ik zal deze keer geen opmerkingen maken over het obsoleete karakter van L^AT_EX 2.09 ☺)

15 Paginalengte veranderen

Bas ter Haar Romeny (*ROMENY@RULLET.LEIDENUNIV.NL*) schreef:

Nou heb ik een probleempje wat vast heel makkelijk op te lossen is, maar waar ik even helemaal niet uitkom. Onderaan het postpapier van de Universiteit waar ik werk staat allerlei adresinformatie, zodat je een ondermarge van zo'n 4 cm moet houden. Op een vervolgvelletje hoeft dat niet, dus daar kan de `\vsize` weer iets groter worden. Als je zelf het begin van een nieuwe pagina kunt/wilt aangeven, is er geen probleem, dan vergroot je de `\vsize` direct na iets als `\newpage`. Maar wat te doen als de eerste pagina bijvoorbeeld midden in een alinea afgebroken wordt?

Piet van oostrum antwoordde:

Ik moet er op wijzen dat L^AT_EX verschillende parameters heeft die ook nog gewijzigd moeten worden.

```
\makeatletter
\newcommand{\addtextheight}[1]
  {\global\advance\textheight#1
  \@colht\textheight
  \@colroom\textheight
  \vsize\textheight}
\makeatother
```

Een goede plaats om `\vsize` te veranderen is in de output routine. Nu is het niet zo'n goed idee om de output routine te veranderen, maar de page headers en footers worden in de output routine veranderd. Je kunt dus bijv. in die standaard footer zoiets zetten:

```
\addtextheight{4cm}
```

Een andere mogelijkheid is om dit met het `afterpage` package te doen.

```
\afterpage{\addtextheight{4cm}}
```

`\colheight` wordt door de output routine zelf al gezet, dus de footer methode kan dit weglaten. Misschien `\colht` ook wel, de timing van de verschillende dingen is lastig te volgen in de L^AT_EX output routine.

16 Raggedright

Werenfried Spit (*w.spit@WITBO.NL*) schreef:

Weet iemand hoe je tijdelijk (bijvoorbeeld binnen een `\parbox`) het commando `\raggedright` kunt aanzetten?

Piet van Oostrum antwoordde:

```
\parbox{\raggedright text.....}
```

Dingen om op te letten:

1. Om het tijdelijk te maken moet je het binnen een groep doen. Dus bijv. tussen `{}` of binnen een omgeving. Het argument van `\parbox` wordt ook al binnen een groep gedaan, dus zijn daar geen extra `{}` nodig.
2. De declaratie geldt voor gehele alinea's. Als je `\raggedright` halverwege een alinea aanzet, dan geldt die voor de hele alinea, zelfs als het deel ervoor buiten de groep staan. In het bijzonder moet gelden dat het alinea-einde (`\par` of lege regel) **in** dezelfde groep staat als de `\raggedright`. Dit komt omdat de situatie bij de `\par` bepalend is voor de layout van de alinea. Je zou dus zelfs kunnen zeggen `\parbox{text..... \raggedright}` maar dat leest contra-intuïtief.
3. `\raggedright` is **zo** ragged dat in de praktijk woorden niet meer afgebroken worden. Er zijn definities in omloop die dat corrigeren. Sommige layouters zullen zeggen dat `\raggedright` niet afgebroken hoort te worden, maar in het nederlands is dat niet echt vol te houden. Zeker niet in smalle kolommen. Een mogelijkheid is om de plain `TEX` definitie te gebruiken:

```
\def\raggedright{\rightskip=0pt plus2em \spaceskip=.3333em
\xspaceskip=.5em\relax}
```

 maar die is niet helemaal compatible met de rest van `LATEX` (wat je alleen zult merken als je binnen deze alinea's weer andere ingewikkelde omgevingen gaat gebruiken). Voor alle zekerheid kun je dan beter geven:

```
\newcommand{\myraggedright}{..}
```

 of zo iets. Er is ook nog een package `ragged2e` waarmee je dingen kunt parametriseren.
4. In een `p{}` kolom van `array/tabular` moet je uitkijken omdat `\raggedright` de `\` herdefinieert. Die heeft dan niet meer de betekenis van een nieuwe kolom, maar een `newline` in de alinea. In de nieuwste `LATEX` is dit opgelost doordat je `\tabularnewline` kunt gebruiken voor de `tabular/array` `\`.

17 Levensverzekeringen

Erik Frambach (*E.H.M.Frambach@eco.rug.nl*) schreef:

In het actuaariaat schijnt men vaker met notaties als deze te werken:
 $\binom{2}{3}A^{\dots 1}_x; n$
 waarbij die "n" nog iets speciaals heeft:

 n | m.a.w. de boven- en rechterkant van een kader.

Hoe maak ik zo'n `\iets {n}`, zodanig dat het met een willekeurig argument (n, k, W, ij, geen ingewikkelde constructies) goed uitziet?

17.1 Eerste antwoord

Herman Haverkort (*hh@FGBBS.IAF.NL*) antwoordde:

```
...
\usepackage{hhflxbox}
...
\def\hoekje#1{%
\sframe[-]\-[1]\sfrlrule[2]\sfrsrule[-]\-{\sepbox(1pt,2pt,2pt,1pt){#1}}
```

(lees: frame met niets links, een liggende lijn boven, een staande lijn rechts, en niets onder, rond een boxje met rond de letter een marge van 1pt links, 2pt boven, 2pt rechts, 1pt onder)

`\hoekje{n}` geeft dan \overline{n}

17.2 Tweede antwoord

Hans Hagen (*pragma@pi.net*) schreef:

Hier een (doodgewone en snelle) plain T_EX oplossing:

```
\def\TRBarSpacing%
  {.1em}
\def\TRBarSpacing%
  {.1em}

\def\TRBar#1%
  {\hbox
    {\lower\TRBarSpacing\hbox
      {\vbox
        {\hrule
          \kern\TRBarSpacing
          \hbox{\kern\TRBarSpacing#1\kern\TRBarSpacing}
          \kern\TRBarSpacing}%
        \vrule}}}}
```

Ziehier een wiskundige $\overline{\text{Actuaris}}$ Actuaris. Als het niet in-line moet, dan volstaat:

```
\def\TRBar#1%
  {\hbox
    {\vbox
      {\hrule
        \kern\TRBarSpacing
        \hbox{\kern\TRBarSpacing#1\kern\TRBarSpacing}
        \kern\TRBarSpacing}%
      \vrule}}
```

Waarbij het geheel boven de baseline wordt geplaatst. Een spacing uitdrukken in em heeft als voordeel dat e.e.a. font afhankelijk is.

17.3 Derde antwoord

Piet Tutelaers (P. T. H. Tutelaers@urc.tue.nl) schreef:

Dit is inderdaad een notatie die gebruikt wordt in de levensverzekeringswiskunde. Ik heb ooit het volgende boek in handen gehad 'Life Insurance Mathematics, second edition' van Hans U. Gerber (1995). Dit boek is met T_EX gemaakt. Ik weet niet of de daarin gebruikte macros beschikbaar zijn. Een letter of cijfer met daarboven een winkelhaak wordt vaak gebruikt voor de looptijd.

Misschien heb je wat aan mijn macros.

```
\documentclass{article}
% symbool voor levensverzekerkunde
\newcommand{\per}[1]{
  \setbox0=\hbox{\$ \overline{\scriptstyle#1} \$}
  \copy0\rule[-0.4pt]{0.4pt}{\ht0}
\newcommand{\ahook}[1]{
  \lower1.2pt\hbox{\scriptstyle#1}}

\begin{document}
\[
a_{\per{x}\,n} = \sum_{r=1}^n A_{\ahook{x}} \cdot \per{r}^{-1}
\]

\[
^2_{\per{x}}A_{\per{x}}^{-1}_n
\]
\end{document}
```

$$a_{\overline{x}|n} = \sum_{r=1}^n A_{x:\overline{r}|}^{-1}$$

$$^2_{\overline{x}}A_{\overline{x}|n}^{-1}$$

18 Blokje

De vragen in dit artikel zijn omkaderd op de volgende manier, door middel van een macro, geschreven door Herman Haverkort:

```
\RequirePackage{tabularx}
\newcommand\blokje[1]{\begin{tabularx}{X}{\hline#1\\\hline}\end{tabularx}}
\newcommand{\vraag}[1]{\blokje{#1}}
```

```
\vraag{Met dank aan allen!}
```

Met dank aan allen!

Conversie van BibTeX naar HTML, TeX en LaTeX

Erik Frambach

Samenvatting

BibTeX-bibliografieën kunnen met behulp van een Perl-script vertaald worden naar HTML. Een variant op dat script levert een compleet LaTeX-bestand. Nog een andere variant levert een plain TeX-bestand dat naar believen bijgeschaafd kan worden. Als gereedschap is enkel Perl en (La)TeX nodig.

Keywords: bibTeX, Perl, HTML, conversie

1 Aanleiding

Ooit is door de MAPS-redactie een begin gemaakt met het opzetten en bijhouden van een MAPS-bibliografie waarin gegevens van alle artikelen die in de MAPS zijn gepubliceerd zijn opgenomen. Helaas is dat mooie initiatief verwaterd, zodat er slechts een schamele en verre van complete versie beschikbaar was. Tijd dus voor een grote inhaalslag waarin alle achterstallige onderhoud wordt uitgevoerd. In die slag zijn meteen alle artikelen uit alle MAPSen als afzonderlijke PDF-bestanden beschikbaar gemaakt (zie ook het artikel over de nieuwe NTG WWW-pagina's elders in deze MAPS).

Uitgaande van de bestaande fragmenten en van alle PDF-versies van de MAPS-artikelen heb ik de MAPS-bibliografie in ruwe vorm gecompleteerd. Vervolgens heb ik de auteurs van artikelen waarin *keywords* en/of *abstract* ontbraken gevraagd die gegevens aan te vullen. Met weinig succes overigens. Daarom nogmaals een oproep: stuur aanvullingen!

Uiteraard moest die MAPS-bibliografie beschikbaar zijn op onze WWW-site, met alle artikelen in PDF die erin genoemd worden. Echter, BibTeX is niet voor iedereen even praktisch. Daarom ben ik gaan zoeken naar mogelijkheden om BibTeX te converteren naar HTML. Echt goed gereedschap heb ik daar niet voor kunnen vinden, dus ben ik eens rond gaan vragen.

Taco Hoekwater stuurde al snel een Perl-script op dat ongeveer deed wat ik wilde. Vervolgens kwam Hans van Mourik met zijn versie. Wybo Dekker heeft toen die twee bewerkt tot één.

Vanuit deze bib2html-converter was het eenvoudig om met enkele aanpassingen ook plain TeX of LaTeX te genereren.

2 Gereedschap

BibTeX-bestanden zijn van nature strak gestructureerd, zodat een 1-op-1 conversie naar HTML/TeX/LaTeX mogelijk moet zijn.

Als gereedschap kwam al snel 'Perl' naar voren, omdat het uitermate geschikt is voor tekstmanipulatie, en omdat het op alle computer-platforms beschikbaar is. Andere opties hebben we niet eens meer onderzocht. Met 'AWK' zou het ook kunnen, of zelfs met TeX, maar met die laatste zou wel erg lastig en bijzonder traag worden.

3 Conversie naar HTML

HTML is redelijk eenvoudig van opzet zodat conversie niet al te moeilijk moet zijn.

Om te beginnen definiëren we het stramien waarbinnen alles gebeurt:

```
<html>
<body>
anything
</body>
</html>
```

'anything' wordt ingevuld als een 'definition list', die als volgt gestructureerd is:

```
<dl>
<dt>term</dt>
<dd>descriptie</dd>
<dt>nog een term</dt>
<dd>nog een descriptie</dd>
etc.
</dl>
```

3.1 Problemen

Het grootste probleem bij conversie naar HTML is het feit dat in HTML geen macro's gedefinieerd kunnen worden. Kortom, alles moet helemaal uitgeschreven worden. Daarmee gaat dus onherroepelijk informatie verloren. Terugconverteren van HTML naar BibTeX is daardoor niet meer zonder meer mogelijk, maar in de regel is dat geen probleem.

Andere problemen zijn meer des TeX's: accenten worden in HTML heel anders geschreven dan in (La)TeX. In HTML zijn bovendien bepaalde accenten die TeX kent niet gedefinieerd, bv. '\n'. Ook kunnen sommige accenten in TeX op verschillende manieren geschreven worden:

'\{\i}' en '\\"i' en '\{\\"i}' zijn equivalent. De convertor moet alle varianten aankunnen.

Speciale tekens zoals '\OE', '\dots', '\l{}' en '\&' vragen ook om een speciale behandeling. De verbatim-omgeving is helemaal een verhaal apart, omdat de eenvoudigste truc die bib2html uithaalt neerkomt op het simpelweg verwijderen van alle '\', '{' en '}'. Dat mag in verbatim-modus natuurlijk niet.

We hebben er bewust voor gekozen mathematische tekens niet te ondersteunen omdat dat te veel problemen geeft. In een bibliografie (zeker die van de MAPS) komen die niet of nauwelijks voor zodat dat geen problemen geeft.

Ook andere constructies moeten vermeden worden in het BibTeX-bestand. Bib2html ondersteunt geen '\bf ...}', '\textit{...}' en dergelijke. Het zal duidelijk zijn dat TeX-macro's definiëren geheid problemen geeft.

4 Conversie naar LaTeX

Bij de conversie naar LaTeX worden de velden uit het BibTeX-bestand overgenomen (met kleine wijzigingen, bv. '\speak' i.p.v. '\language'). Voor het zetten van die velden worden macro's gedefinieerd, zodat het LaTeX-bestand in één keer gecompileerd kan worden en meteen fraai geformateerde uitvoer levert.

5 Conversie naar TeX

De opzet is hier in principe dezelfde als bij conversie naar LaTeX, maar de macro's voor opmaak zijn 'leeg' gelaten. De uitvoer zal in eerste instantie niet fraai zijn. De TeX-uitvoer is meer bedoeld voor specialistische (database-achtige) bewerkingen zoals Hans Hagen die demonstreert in <http://www.ntg.nl/maps/pdf/maps.pdf>.

Dat document is gegenereerd met de naar TeX geconverteerde MAPS-bibliografie als invoer. De uitvoer is een interactief document waarin door de bibliografie gewandeld kan worden via verschillende 'linked-lists' of dimensies. Voorbeelden: alle artikelen van één MAPS op een rijtje; alle artikelen van één auteur op een rijtje; alle artikelen met een bepaald *keyword* op een rijtje. Uiteraard zijn er ook verschillende indexen beschikbaar. En dit alles wordt volautomatisch gegenereerd!

6 Voorbeelden

Als demonstratie volgt hier een klein stukje BibTeX-code, met daarna de uitvoer van de conversie naar HTML, plain TeX en LaTeX. De BibTeX-code is geplukt uit de MAPS-bibliografie en is 'verrijkt' met enkele grapjes om te laten zien hoe bib2html met 'moeilijke' gevallen omgaat.

```
@ARTICLE{2-6,
  author = {{Theo de Klerk}},
  names = {klerktde},
  title = {{Boeken over \TeX}},
```

```
  year = {{1989}},
  language = {{Dutch}},
  journal = {{MAPS}},
  volume = {{2}},
  pages = {{19-20}},
  size = {{71}},
  keywords = {{boekbespreking \& \OE{}uvre,
    \verb+\def\TeX{difficult}+}},
  abstract = {{Bespreki\'ng 'Einf\'uhrung in
    \TeX\dots' (Norbert Schwartz); '\TeX f\'ur
    Fortgeschrittene' (Wolfgang Appe\l{}t); \
    '\LaTeX\ eine Einf\'uhrung' (Helmut Kopka);
    'Kompaktf\'uhrer \LaTeX' (Reinhard Wonneberger)}}
}
```

Alle velden zijn voorzien van dubbele accolades. De reden hiervoor is dat ik in dit geval niet wil dat BibTeX zelf met hoofdletters en kleine letters gaat stoeien. Meer gebruikelijk is het om in zo'n geval bv. te schrijven 'Dit {I}s {\TeX}'. Dan zal de hoofdletter 'I' beslist een hoofdletter blijven (ongeacht de voorschriften in de BibTeX-style) en zal de TeX-compiler niet gaan klagen dat ie het commando '\tex' niet kent. Omdat dit een erg onhandige manier van schrijven vereist heb ik ervoor gekozen alles af te schermen, maar toch BibTeX-compatibel te blijven.

Het veld 'names' is uitsluitend bedoeld als sorteersleutel en wordt in de regel niet afgedrukt.

6.1 HTML

(De formattering is iets aangepast om de code hier netjes te kunnen afdrukken. Regelovergangen hebben echter voor HTML dezelfde betekenis als spaties, dus hindert dat niet.)

```
<html>
<body>
<dl>
<dt class="ARTICLE" id="2-6"><strong>Theo
  de Klerk</strong></dt>
<dd>
  <em>Boeken over TeX</em>, Dutch, MAPS
  <strong>2</strong>, 1989, pp. 19-20<br>
  <strong>keywords:</strong> boekbespreking &
  Oeuvre, <tt>\def\TeX{difficult}</tt><br>
  <strong>abstract:</strong> Bespreking
  'Einf&uuml;hrung in TeX...' (Norbert Schwartz);
  'TeX f&uuml;r Fortgeschrittene' (Wolfgang Appelt);<br>
  'LaTeX eine Einf&uuml;hrung' (Helmut Kopka);
  'Kompaktf&uuml;hrer LaTeX' (Reinhard Wonneberger)<br>
  <a href="pdf/2_6.pdf">2-6 in PDF</a> (71 KB)
</dd>
</dl>
</body>
</html>
```

6.2 LaTeX

```
\documentclass[a4paper]{article}

% macros:
```

```

\newif\ifseteditor
\def\editstring{}
\def\startentry#1{%
  \def\id{#1}
  \seteditortrue\par\noindent\hangindent 10mm}
\def\author#1{%
  \seteditorfalse#1\}
\def\names#1{}
\def\editor#1{%
  \ifseteditor
    #1 (editor)\
  \else
    \def\editstring{#1 (editor)}
  \fi}
\def\title{%
  \begingroup\bf}
\def\speak#1{%
  \endgroup
  \ #1,}
\def\journal#1{%
  {\sl#1}}
\def\volume#1{%
  {\bf#1},}
\def\series#1{%
  #1,}
\def\year#1{%
  #1,}
\def\booktitle#1{%
  in:~{\sl#1},}
\def\publisher#1{%
  \editstring, #1,}
\def\pages{%
  pp.~}
\def\keywords{
  \ \bf keywords: }}
\def\abstract{%
  \ \bf abstract: }}
\def\size#1{%
  (\id.pdf: #1 KB)}
\def\stopentry{}

% end of macros

\begin{document}

\startentry {2-6}
\author {Theo de Klerk}
\names {klerktde}
\title {Boeken over \TeX}
\speak {Dutch}
\journal {MAPS}
\volume {2}
\year {1989}
\pages {19-20}
\keywords {boekbespreking \& \OE{}uvre,
  \verb+\def\TeX{difficult}+}
\abstract {Bespreki'ng 'Einf\"uhrung in \TeX\dots'
  (Norbert Schwartz); '\TeX f\"ur Fortgeschrittene'
  (Wolfgang Appe\l{}t);\
  '\LaTeX\ eine Einf\"uhrung' (Helmut Kopka);
  'Kompaktf\"uhrer \LaTeX' (Reinhard Wonneberger)}
\size {71}
\stopentry

\end{document}

```

6.3 plain TeX

```

\def\LaTeX{LaTeX}
\def\{\hfил\break}

\def\startentry{}
\def\author{}
\def\names{}
\def\editor{}
\def\title{}
\def\speak{}
\def\journal{}
\def\volume{}
\def\series{}
\def\year{}
\def\booktitle{}
\def\publisher{}
\def\pages{}
\def\keywords{}
\def\abstract{}
\def\size{}
\def\stopentry{}

\def\type{} % !!! macro for typesetting
              % !!! verbatim text

\startentry {2-6}
\author {Theo de Klerk}
\names {klerktde}
\title {Boeken over \TeX}
\speak {Dutch}
\journal {MAPS}
\volume {2}
\year {1989}
\pages {19-20}
\keywords {boekbespreking \& \OE{}uvre,
  \type{\def\TeX{difficult}}}
\abstract {Bespreki'ng 'Einf\"uhrung in \TeX\dots'
  (Norbert Schwartz); '\TeX f\"ur Fortgeschrittene'
  (Wolfgang Appe\l{}t);\
  '\LaTeX\ eine Einf\"uhrung' (Helmut Kopka);
  'Kompaktf\"uhrer \LaTeX' (Reinhard Wonneberger)}
\size {71}
\stopentry
\bye

```

7 Conclusie

BibTeX-documenten zijn strak gestructureerd. Dit maakt het relatief eenvoudig om ze te converteren naar andere systemen die gestructureerd werken mogelijk maken. LaTeX ligt daarbij voor de hand, maar ook plain TeX doet het prima en geeft uiteindelijk zelfs meer flexibiliteit, zij het dat daarvoor navenant meer programmeerwerk nodig is.

8 Beschikbaarheid

De converters zullen binnenkort op CTAN gezet worden.

Het invoeren en afdrukken van de Latin-1 (ISO-8859-1) Characterset

Ton Biegstraaten

Samenvatting

Ten behoeve van een collegedictaat is een tabel nodig met de Latin-1 of ISO-8859-1 characterset. Dit artikel behandelt het invoeren en afdrukken van deze characterset met behulp van L^AT_EX₂ε.

1 Enige achtergrond informatie

1.1 T_EX

Sinds T_EX versie 3 kunnen naast de gebruikelijke charactersets met 128 tekens ook sets met 256 tekens gedefinieerd en gebruikt worden. Een voorbeeld is de ISO-8859-1 of Latin-1 characterset welke voldoende symbolen bevat om de meeste Europese talen te kunnen verwerken.

1.2 Extended Computer Modern fonts

In 1990 gedurende de TUG conferentie in Cork, Ierland is een nieuw font coderingsschema vastgesteld welke veel Europese talen met een Latijns schrift ondersteunt. De eerste 128 posities van dit schema komen overeen met de bestaande codering voor de computer modern fonts. De laatste 128 posities worden ingenomen door characters met accenten en bijzondere taalspecifieke characters. Voorheen konden sommige van deze characters alleen als combinatie van andere characters worden afgedrukt. Anderen konden niet worden afgedrukt.

Omdat verschillende fonts verschillende coderingsschema's kunnen hebben, moeten ze onderscheiden kunnen worden. Ze krijgen daarom een unieke naam. De nieuwe codering heeft de naam: T1, de oude OT1, old of obsolete T1. Zo zijn er meerdere coderingsschema's gedefinieerd.

Aan een codering alleen heb je niet veel, er horen fonts bij waardoor de codering te gebruiken is. Deze fonts zijn begin 1997 opgeleverd en worden 'extended computer modern fonts' of ec fonts genoemd. Tijdens de ontwikkeling zijn ze beschikbaar geweest als dc fonts. Alle voorbeelden in dit artikel werken zowel met de dc als met de ec fonts.

Naast de ec fonts was er behoefte aan fonts met symbolen die voorheen alleen in mathematische fonts beschikbaar waren. Deze fonts worden de 'text companion' fonts genoemd (tc fonts). Hierin kunnen zich ook symbolen bevinden die geen plaats hebben in de ec fonts.

2 Het gebruik van de ISO-8859-1 set

Wanneer bepaalde characters moeten worden afgedrukt kunnen ze in fonttabellen opgezocht worden. Wanneer

ze gevonden zijn, kunnen ze expliciet opgenomen worden door het betreffende font te definiëren. Incidenteel is deze methode heel goed bruikbaar, maar voor een gestandaardiseerde characterset als de ISO-8859-1 set zou een meer algemene benadering gewenst zijn.

Het blijkt dat de genoemde ec en tc fonts alle characters bevatten om de ISO-8859-1 set volledig te kunnen afdrukken. De volgende vragen blijven dan nog over:

1. Hoe voer je ISO-8859-1 characters in in je computer.
2. Hoe druk je ze via T_EX op een zo elegant mogelijke manier af.

2.1 Het invoeren van ISO-8859-1 characters

Deze sectie is noodzakelijkerwijs nogal systeemafhankelijk. Ik werk voornamelijk onder Linux met emacs en vi als editors.

Een deel van de ISO-8859-1 characters, die met accenten, kunnen natuurlijk op de gebruikelijke manier worden opgegeven, é kan nog steeds worden ingevoerd als \ ' e. Dit gaat niet voor alle characters en er moeten dus andere manieren bestaan.

Een standaard toetsenbord kent alleen de mogelijkheid de gewone ASCII characters in te toetsen. Dit is slechts een subset van de ISO standaard. Characters met ASCII code > 127 kunnen niet direct worden opgegeven. Het achterliggende computersysteem zou er dan voor kunnen zorgen dat bepaalde combinaties van tekens worden omgezet in het gewenste character. Hiervoor zijn verscheidene mogelijkheden:

- Onder Linux (en ongetwijfeld onder meer operating systems) kan gebruik gemaakt worden van de Alt toets, waarbij meestal eerst gezorgd moet worden dat de Alt toets niet wordt afgevangen door een shell voor andere doeleinden. Hiervoor moet onder Linux C-v (de Ctrl toets tesamen met de v toets) gegeven worden. De ASCII waarde van de toets die tegelijk wordt ingedrukt met de Alt toets wordt met 128 verhoogd. Onder openwin kan dit problemen geven. Alt-q b.v. gooit (onder Linux) het window weg (ook na C-v). Het is mij nog niet gelukt dit uit te zetten. Ik heb niet nog

meer van dit soort combinaties gevonden, maar je weet maar nooit.

Verder is er geen relatie tussen het ASCII character en het character met een waarde 128 groter. B.v. Alt-" geeft \242 in een xterm-window. In vi wordt dit ¢. De characters hebben duidelijk geen relatie tot elkaar. In vi kan de Alt combinatie overigens direct worden ingetoetst, zonder C-v ervoor.

Combinaties als Alt-q kunnen (in ieder geval onder Linux) niet worden gegeven, en het bijbehorende symbool is dan niet bruikbaar. Bij gebruik van twm of fvwmm in plaats van openwin speelt dit probleem niet.

- In emacs zijn een aantal bibliotheek functies aanwezig die het gebruik van ISO-8859-1 vereenvoudigen. Deze zijn vanaf emacs 19 beschikbaar gekomen.

De standaard methode is het geven van C-q gevolgd door drie octale cijfers die gelijk zijn aan de charactercode. Het nadeel is dat deze code bekend moet zijn en er geen herkenbare relatie bestaat tussen de code en het character.

De eerder genoemde methode, waarbij de Alt toets wordt gebruikt, is binnen emacs ook mogelijk. Deze toets is hier ook in gebruik voor andere functies en moet daarom afgeschermd worden. Dit kan door eerst C-q te geven. Ook hier kunnen problemen ontstaan doordat characters door het omringende systeem worden afgevangen, zoals Alt-q.

De twee genoemde methoden hebben allebei nadelen. Er is nog een derde invoermethode mogelijk welke gebruik gemaakt van een prefix code n.l.: C-x8 gevolgd door twee characters die tesamen zo veel mogelijk een indicatie zijn van het nieuwe te vormen character. ¢ ontstaat nu door C-x8*c te geven. Deze mogelijkheid moet via een bibliotheek functie worden opgestart. Dit gaat met:

```
Alt-x load-library ;enter; iso-transl@.
```

Standaard wordt na ingave van een niet ASCII character de octale representatie gegeven. Na installeren van de iso-ascii bibliotheek functie (Alt-x load-library ;enter; iso-ascii@) wordt tussen accolades een indicatie van het teken gegeven. ¢ wordt dan weergegeven als {c}.

Na Alt-x standard-display-european wordt het werkelijke teken weergegeven, dit werkt niet altijd, het omringende systeem (b.v. X-Windows) moet dit ondersteunen.

2.2 Alle gegevens bij elkaar

In de tabel aan het einde van dit artikel, worden alle genoemde mogelijkheden op een rij gezet. Zowel het character, als de octale, decimale en hexadecimale representatie worden gegeven. Daarnaast wordt gegeven welke de toets is die tesamen met de Alt toets moet worden ingedrukt om het betreffende character te krijgen. Ook de emacs toets combinatie die na C-x8 gegeven moet worden wordt

vermeld. In de laatste kolom wordt de ASCII representatie gegeven die binnen emacs gebruikt kan worden op een scherm waar de echte symbolen niet aanwezig zijn.

Alleen de niet-ASCII characters worden weergegeven, de ASCII characters zelf wijzigen niet wat betreft invoer en weergave.

Wanneer er achter een character (m) staat, is het character alleen in mathmode beschikbaar en moet binnen textmode tussen twee \$'s worden geplaatst worden. Deze characters zijn ook in textmode beschikbaar via de tc fonts. Hierop kom ik in de volgende sectie terug.

2.3 Het gebruik binnen L^AT_EX₂ε

De wijze waarop de volledige ISO-8859-1 characterset kan worden ingevoerd is in de vorige subsectie duidelijk gemaakt. De vraag die nu nog rest is of T_EX daar iets mee kan. Gezien de beloofde tabel zal dat wel vermoed ik.

Bij de bespreking van de ec fonts is aangegeven dat het font coderingsschema de naam T1 heeft. In L^AT_EX₂ε is het mogelijk dit op te geven door T1 op te geven als parameter aan de style file fontenc.d.m.v.

```
\usepackage[T1]{fontenc}.
```

Hierdoor wordt automatisch van de dc fonts gebruik gemaakt.¹ Echter een character positie in de dc fonttabel komt i.h.a. niet overeen met haar positie in de ISO-8859-1 characterset. Er moet dus nog een mapping tussen de ISO-set en de ec fonttabel gemaakt worden.

Dit kan gebeuren door naast de fontcodering ook de inputcodering op te geven. D.m.v. de style file inputenc is dit mogelijk. Deze heeft een parameter latin1, synoniem voor ISO-8859-1. Hiermee lijkt het probleem opgelost.

Dit blijkt echter niet het geval te zijn. Er zijn vier characters die wel in de ISO-8859-1 characterset voorkomen maar niet in de ec fonts. Ze zijn echter wél in de al eerder genoemde tc fonts aanwezig, maar worden niet standaard gebruikt. Na het meegeven van de style file textcomp gebeurt dit wél. Deze style file bevat nog veel meer definities welke symbolen die alleen in mathmode beschikbaar zijn in text mode beschikbaar maken. Dit kan o.a. zinvol zijn voor verbatim listings van symbolen. Omdat een symbool óf in math mode óf in textmode beschikbaar is kan dit ongewenste effecten hebben voor de rest van een artikel.

Jörg Knappen de persoon die de ec en tc fonts onderhoudt heeft speciaal voor verbatim listings van symbolen een speciale versie van latin1 gemaakt, genaamd latin1jk. Wanneer deze in dit artikel zou worden gebruikt kunnen alle symbolen uit de tabel die als mathematisch vermeld staan worden ontdaan van hun dollars omdat ze nu in textmode beschikbaar zijn. Na navraag bij hem bleek dat hij deze speciale versie blijft onderhouden en dat er dus algemeen gebruik van kan worden gemaakt.

De preamble van de L^AT_EX file welke op dit moment het gewenste resultaat geeft ziet er als volgt uit:

¹Wanneer de ec fonts aanwezig zijn kan standaard voor deze fonts worden gekozen. Zie hiervoor de installatie beschrijving bij de ec fonts.

```
\documentclass[11pt]{artikel13}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
```

```
\usepackage[latin1]{inputenc}
\usepackage[dutch]{babel}
```

Nu volgt de eerder genoemde tabel met alle gegevens:

Char	Oct	Dec	Hex	C-x 8	Alt-	ISO-ASCII
	240	160	A0	* \square	\square	
ı	241	161	A1	*!	!	{!}
ç	242	162	A2	*c	"	{c}
£	243	163	A3	*L	#	{GBP}
¤	244	164	A4	*\$	\$	{\$}
¥	245	165	A5	*Y	%	{JPY}
¦	246	166	A6	*	&	{ }
§	247	167	A7	*S	'	{S}
¨	250	168	A8	""	({"}
©	251	169	A9	*C)	{C}
^a (m)	252	170	AA	_a	*	{_a}
«	253	171	AB	*<	+	{<<}
¬ (m)	254	172	AC	~~	,	{~}
	255	173	AD	*-	-	{-}
®	256	174	AE	*R	.	{R}
-	257	175	AF	*=	/	{=}
° (m)	260	176	B0	*o	0	{o}
± (m)	261	177	B1	*+	1	{+-}
² (m)	262	178	B2	^2	2	{2}
³ (m)	263	179	B3	^3	3	{3}
'	264	180	B4	'	4	{'}
μ (m)	265	181	B5	*u	5	{u}
¶	266	182	B6	*P	6	{P}
·	267	183	B7	*.	7	{.}
¸	270	184	B8	,,	8	{,}
¹ (m)	271	185	B9	^1	9	{1}
º (m)	272	186	BA	_o	:	{_o}
»	273	187	BB	*>	;	{>>}
¼	274	188	BC	1/4	<	{1/4}
½	275	189	BD	1/2	=	{1/2}
¾	276	190	BE	3/4	>	{3/4}
¿	277	191	BF	*?	?	{?}
À	300	192	C0	'A	@	{'A}
Á	301	193	C1	'A	A	{'A}
Â	302	194	C2	^A	B	{^A}
Ã	303	195	C3	~A	C	{~A}
Ä	304	196	C4	"A	D	{"A}
Å	305	197	C5	/A	E	{AA}
Æ	306	198	C6	/E	F	{AE}
Ç	307	199	C7	,C	G	{,C}
È	310	200	C8	'E	H	{'E}
É	311	201	C9	'E	I	{'E}
Ê	312	202	CA	^E	J	{^E}
Ë	313	203	CB	"E	K	{"E}
Ì	314	204	CC	'I	L	{'I}
Í	315	205	CD	'I	M	{'I}
Î	316	206	CE	^I	N	{^I}
Ï	317	207	CF	"I	O	{"I}
	320	208	D0	~D	P	{~D}
Ñ	321	209	D1	~N	Q	{~N}
Ò	322	210	D2	'O	R	{'O}

Char	Oct	Dec	Hex	C-x 8	Alt-	ISO-ASCII
Ó	323	211	D3	'O	S	{'O}
Ô	324	212	D4	^O	T	{^O}
Õ	325	213	D5	~O	U	{~O}
Ö	326	214	D6	"O	V	{"O}
× (m)	327	215	D7	*x	W	{x}
Ø	330	216	D8	/O	X	{/O}
Ù	331	217	D9	'U	Y	{'U}
Ú	332	218	DA	'U	Z	{'U}
Û	333	219	DB	^U	[{^U}
Ü	334	220	DC	"U	\	{"U}
Ý	335	221	DD	'Y]	{'Y}
	336	222	DE	~T	^	{TH}
ß	337	223	DF	"s	_	{ss}
à	340	224	E0	'a	'	{'a}
á	341	225	E1	'a	a	{'a}
â	342	226	E2	^a	b	{^a}
ã	343	227	E3	~a	c	{~a}
ä	344	228	E4	"a	d	{"a}
å	345	229	E5	/a	e	{aa}
æ	346	230	E6	/e	f	{ae}
ç	347	231	E7	,c	g	{,c}
è	350	232	E8	'e	h	{'e}
é	351	233	E9	'e	i	{'e}
ê	352	234	EA	^e	j	{^e}
ë	353	235	EB	"e	k	{"e}
ì	354	236	EC	'i	l	{'i}
í	355	237	ED	'i	m	{'i}
î	356	238	EE	^i	n	{^i}
ï	357	239	EF	"i	o	{"i}
	360	240	F0	~d	p	{-d}
ñ	361	241	F1	~n	q	{~n}
ò	362	242	F2	'o	r	{'o}
ó	363	243	F3	'o	s	{'o}
ô	364	244	F4	^o	t	{^o}
õ	365	245	F5	~o	u	{~o}
ö	366	246	F6	"o	v	{"o}
÷ (m)	367	247	F7	//	w	{/}
ø	370	248	F8	/o	x	{/o}
ù	371	249	F9	'u	y	{'u}
ú	372	250	FA	'u	z	{'u}
û	373	251	FB	^u	{	{^u}
ü	374	252	FC	"u		{^"u}
ý	375	253	FD	'y	}	{'y}
	376	254	FE	~t	~	{th}
ÿ	377	255	FF	"y	DEL	{"y}

SGML entities in (L^A)T_EX: sgmlent.sty

T. Hoekwater

Abstract

This package provides an easy-to-use interface to the SGML character entity sets (ISO 8879, Annex D). All entity sets defined in the Annex are supported, but there are still some minor problems.

Keywords: SGML entities, structural SGML entity conversions, SGML -> T_EX

1 Overview

This package requires the packages/distributions `amsmath`, `wasysym` and quite some fonts, of which most are included with the package `sgmlent`.

All of the entities defined in Annex D of the standard are given a 'standardized' L^AT_EX name. This is intended to make SGML \implies L^AT_EX conversion easier. The final sections of this article list all the entities, together with their graphic representation, and some technical notes where needed.

2 Naming of entities

The naming scheme that is use for the T_EX macros is slightly verbose, but more importantly, it is consistent. Conversion from the names used in the annex to the L^AT_EX commands can be done using only four simple rules:

1. **All entity names are prefixed with `\sgm`**
2. **All macro names end with `;` (semicolon).**
This makes it easier to maintain correct spacing, since T_EX will not ignore space characters after the `;`. Although this means the macros break for instance if you use letterspaced text, in my opinion it is more important that the conversions I write can completely ignore spacing around entities.
3. **In the SGML names, all numbers are converted to english words.**
Note that this is single character-based, so that '1' becomes 'one', '6' becomes 'six' but '90' becomes 'ninezero'.
4. **A dot in the entity name (as in the "Alternative Greek Symbols") is replaced with a "d".**

Here are some examples:

```
&blk14; becomes \sgmblkonefour;
&Aacute; becomes \sgmAacute;
&b.eta; becomes \sgmbdeta;
&darr2; becomes \sgmdarrtwo;
```

This is done throughout. You should always be able to generate the T_EX name from the entity name quite easily.

3 Usability constraints

Although this package will also run under T_EX and L^AT_EX209, best results are obtained by using L^AT_EX2_ε with T1 encoding on a PostScript output device.

4 Requirements

4.1 Required extra packages/macro files

The functionality of the 'amsmath' and 'wasysym' packages (resp. `amsmath.tex` and `wasysym.tex` for T_EX) is needed. Because of the size of the corresponding fonts, these are not contained within the distribution of this style-file.

For now, the package depends on the definitions made by those packages. The need for the `amstex` macros may change in the future, but the dependency on the AMS fonts will definitely remain.

4.2 Required fonts

Following the need for 'amsmath' and 'wasysym', you will also need to have the corresponding fonts installed. The fontfamily "wncyr" is included with the `amsmath` distribution, and is used for the Cyrillic entities.

The entities `\sgmxcross;` (X), `\sgmstar;` (☆) and `\sgmsex;` (★) depend on the PostScript font `ZapfDingbats`. Metrics are included in the distribution of this package.

Unfortunately, there is no PostScript Type 1 font for the Waldi symbol font, but maybe it is possible to find different characters in other fonts and merge them together in the future. At the moment, I do not have enough spare time to investigate this option.

The monotoniko greek entities (`grk1` and `grk2`) use the `mrg` fonts by Yannis Haramboulos.

5 What needs to be done yet

- More testing.
- Less dependencies on external macros.

- A merged font would be nice, instead of borrowing from 6 different symbol fonts.
- Better behaviour under non-T1 font encodings.
- Option to bind the correct entities to inputenc.
- Some characters should be changed to look better.

6 Known problems and shortcomings

There are problems (as usual) under T_EX and L^AT_EX209 with respect to sizes and font styles for the characters that are not symbols.

PostScript fonts typically do not contain the glyphs for eng, ENG and dotlessj (and some other also, but these three bother us).

7 Full list of control sequences and their graphical representation

7.1 iso-amsa.def

\curvearrowright	<code>\sgmcularr;</code>	\leftrightarrow	<code>\sgmharr;</code>	\nrightarrow	<code>\sgmnrArr;</code>	\swarrow	<code>\sgmdlarr;</code>
\curvearrowleft	<code>\sgmcurarr;</code>	\Uparrow	<code>\sgmlrarrtwo;</code>	\nearrow	<code>\sgmnwarr;</code>	\Uparrow	<code>\sgmuArr;</code>
\Downarrow	<code>\sgmdArr;</code>	\Uparrow	<code>\sgmlrarrtwo;</code>	\circlearrowleft	<code>\sgmolarr;</code>	\Uparrow	<code>\sgmuarrtwo;</code>
\Downarrow	<code>\sgmdarrtwo;</code>	\rightsquigarrow	<code>\sgmharrw;</code>	\circlearrowright	<code>\sgmorarr;</code>	\Updownarrow	<code>\sgmvArr;</code>
\downarrow	<code>\sgmdharl;</code>	\Rightarrow	<code>\sgmlharttwo;</code>	\Rightarrow	<code>\sgmrAarr;</code>	\Updownarrow	<code>\sgmvarr;</code>
\downarrow	<code>\sgmdharr;</code>	\Rightarrow	<code>\sgmlrharttwo;</code>	\rightarrow	<code>\sgmRarr;</code>	\uparrow	<code>\sgmuharl;</code>
\Leftarrow	<code>\sgmlAarr;</code>	\uparrow	<code>\sgmlsh;</code>	\Rightarrow	<code>\sgmrarrtwo;</code>	\uparrow	<code>\sgmuharr;</code>
\leftarrow	<code>\sgmLarr;</code>	\mapsto	<code>\sgmmap;</code>	\rightarrow	<code>\sgmrarrhk;</code>	\Leftarrow	<code>\sgmxlArr;</code>
\Leftarrow	<code>\sgmlarrtwo;</code>	\circ	<code>\sgmmumap;</code>	\rightarrow	<code>\sgmrarrlp;</code>	\Leftrightarrow	<code>\sgmxhArr;</code>
\leftarrow	<code>\sgmlarrhk;</code>	\nearrow	<code>\sgmnearr;</code>	\rightarrow	<code>\sgmrarrtl;</code>	\leftrightarrow	<code>\sgmxharr;</code>
\rightarrow	<code>\sgmlarrlp;</code>	\nrightarrow	<code>\sgmnlArr;</code>	\rightsquigarrow	<code>\sgmrarrw;</code>	\Rightarrow	<code>\sgmxrArr;</code>
\rightarrow	<code>\sgmlarrtl;</code>	\nrightarrow	<code>\sgmnlarr;</code>	\rightarrow	<code>\sgmrhard;</code>		
\dashrightarrow	<code>\sgmlhard;</code>	\nrightarrow	<code>\sgmnhArr;</code>	\rightarrow	<code>\sgmrharu;</code>		
\dashrightarrow	<code>\sgmlharu;</code>	\nrightarrow	<code>\sgmnharr;</code>	\rightarrow	<code>\sgmrsh;</code>		
\nrightarrow	<code>\sgmhArr;</code>	\nrightarrow	<code>\sgmnrarr;</code>	\searrow	<code>\sgmdrarr;</code>		

7.2 iso-amsb.def

I am quite unhappy with shape of the `\sgmBarwed`; ($\overline{\wedge}$) and `\sgmbarwed`; ($\overline{\wedge}$) entities.

\amalg	<code>\sgmamalg;</code>	\times	<code>\sgmltimes;</code>	\dagger	<code>\sgmplusdo;</code>	\top	<code>\sgmtop;</code>
$\overline{\wedge}$	<code>\sgmBarwed;</code>	\square	<code>\sgmminusb;</code>	\sphericalangle	<code>\sgmrthree;</code>	\oplus	<code>\sgmuplus;</code>
$\overline{\wedge}$	<code>\sgmbarwed;</code>	\otimes	<code>\sgmoast;</code>	\times	<code>\sgmrtime;</code>	\wr	<code>\sgmwreath;</code>
\cap	<code>\sgmCap;</code>	\odot	<code>\sgmocir;</code>	\cdot	<code>\sgmsdot;</code>	\circ	<code>\sgmxcirc;</code>
\cup	<code>\sgmCup;</code>	\ominus	<code>\sgmodash;</code>	\square	<code>\sgmsdotb;</code>	∇	<code>\sgmxdtri;</code>
\vee	<code>\sgmcuvee;</code>	\odot	<code>\sgmodot;</code>	\setminus	<code>\sgmsetmn;</code>	\triangle	<code>\sgmxutri;</code>
\wedge	<code>\sgmcuwed;</code>	\ominus	<code>\sgmominus;</code>	\sqcap	<code>\sgmsqcap;</code>	\amalg	<code>\sgmcoprod;</code>
\diamond	<code>\sgmdiam;</code>	\oplus	<code>\sgmoplus;</code>	\sqcup	<code>\sgmsqcup;</code>	\prod	<code>\sgmprod;</code>
$*$	<code>\sgmdivonx;</code>	\oslash	<code>\sgmosol;</code>	\setminus	<code>\sgmssetmn;</code>	\sum	<code>\sgmsum;</code>
\top	<code>\sgmintcal;</code>	\otimes	<code>\sgmotimes;</code>	\star	<code>\sgmsstarf;</code>		
λ	<code>\sgmlthree;</code>	\boxplus	<code>\sgmplusb;</code>	\boxtimes	<code>\sgmtimeb;</code>		

7.3 iso-amsc.def

\lceil	<code>\sgmrceil;</code>	\lrcorner	<code>\sgmurcorn;</code>	\lfloor	<code>\sgmlfloor;</code>	\llcorner	<code>\sgmdlcorn;</code>
\rfloor	<code>\sgmrfloor;</code>	\lrcorner	<code>\sgmdrcorn;</code>	$\not\propto$	<code>\sgmlpargt;</code>		
\gtr	<code>\sgmrpargt;</code>	\lceil	<code>\sgmlceil;</code>	\ulcorner	<code>\sgmulcorn;</code>		

7.4 iso-amsn.def

∇	<code>\sgmgnap;</code>	∇	<code>\sgmgvnE;</code>	∇	<code>\sgmlnsim;</code>	\neq	<code>\sgmnequiv;</code>
∇	<code>\sgmgne;</code>	∇	<code>\sgmlnap;</code>	∇	<code>\sgmlvnE;</code>	\neq	<code>\sgmngE;</code>
∇	<code>\sgmgnE;</code>	∇	<code>\sgmlnE;</code>	∇	<code>\sgmnap;</code>	\neq	<code>\sgmngE;</code>
∇	<code>\sgmgnsim;</code>	∇	<code>\sgmlne;</code>	∇	<code>\sgmncong;</code>	\neq	<code>\sgmnges;</code>

$\not\approx$ \sgmngt;	∇ \sgmnrtri;	$\not\approx$ \sgmnsup;	$\not\approx$ \sgmscne;
∇ \sgmnlE;	∇ \sgmnrtrie;	∇ \sgmnsupE;	∇ \sgmscnsim;
∇ \sgmnlE;	∇ \sgmnsc;	∇ \sgmnsupe;	∇ \sgmsubne;
∇ \sgmnlE;	∇ \sgmnscE;	∇ \sgmnvdash;	∇ \sgmsubnE;
∇ \sgmnlE;	∇ \sgmnsim;	∇ \sgmnvDash;	∇ \sgmsupne;
∇ \sgmnltri;	∇ \sgmnsime;	∇ \sgmnVDash;	∇ \sgmsupnE;
∇ \sgmnltrie;	∇ \sgmnsimid;	∇ \sgmnVdash;	∇ \sgmvsubnE;
∇ \sgmnmid;	∇ \sgmnspar;	∇ \sgmprnap;	∇ \sgmvsubne;
∇ \sgmnpa;	∇ \sgmnsub;	∇ \sgmprnE;	∇ \sgmvsupne;
∇ \sgmnp;	∇ \sgmnsube;	∇ \sgmprnsim;	∇ \sgmvsupnE;
∇ \sgmnp;	∇ \sgmnsubE;	∇ \sgmscnap;	

7.5 iso-amso.def

There are a few problems with `\sgminodot`; (*i*) and `\sgmjnodot`; (*j*). At the moment, I allow only the math versions of these two entities, because most PostScript font do not even have *j*, but the problem is internal to the macros: I typesets either the text or the math version, and cannot decide which to use. This is a bug and should be fixed asap.

\angle \sgmang;	\top \sgmdaleth;	ι \sgminodot;	\Re \sgmreal;
\angle \sgmangmsd;	ℓ \sgmell;	j \sgmjnodot;	\backslash \sgmsbsol;
\sqcap \sgmbeth;	\emptyset \sgmempty;	$\#$ \sgmnexist;	\prime \sgmvprime;
\backslash \sgmbprime;	\beth \sgmgimel;	\textcircled{S} \sgmoS;	\wp \sgmweierp;
\mathcal{C} \sgmcomp;	\mathfrak{C} \sgmimage;	\hbar \sgmplanck;	

7.6 iso-amr.def

`\sgmecolon`; (=), `\sgmcolone`; (:=) and `\sgmveebar`; (\vee) look silly. And all waves from the AMS fonts look a bit childish, but i am afraid to change those, because it would create two new fonts and even more confusion.

\approx \sgmape;	\equiv \sgmerDot;	\lessdot \sgmlsim;	\sqsupset \sgmsqsup;
\asymp \sgmasymp;	\pitchfork \sgmfork;	\ll \sgmLt;	\sqsupseteq \sgmsqsupe;
\cong \sgmbcong;	\frown \sgmfrown;	\triangleleft \sgmltrie;	\smile \sgmssmile;
\bumpeq \sgmbepsi;	\gap \sgmgap;	\mid \sgmmid;	\Subset \sgmSub;
\bowtie \sgmbowtie;	\gtdot \sgmgdsdot;	\models \sgmmodels;	\supseteq \sgmsubE;
\sim \sgmbsim;	\gE \sgmgE;	\r \sgmpr;	\supset \sgmSup;
\simeq \sgmbsime;	\gEl \sgmgEl;	\r \sgmprap;	\supseteq \sgmsupE;
\bumpeq \sgmbump;	\gEl \sgmgEl;	\r \sgmprpre;	\thkap \sgmthkap;
\bumpeq \sgmbumpe;	\gges \sgmgges;	\r \sgmprsim;	\thksim \sgmthksim;
\circ \sgmcire;	\gg \sgmGg;	∇ \sgmrtrie;	\triangle \sgmtrie;
\colone \sgmcolone;	\gg \sgmgL;	\sqcap \sgmsamalg;	\textcircled{X} \sgmtwixt;
\cuepr \sgmcuepr;	\gg \sgmgS;	\smile \sgmsc;	\vdash \sgmvdash;
\cuesc \sgmcuesc;	\gg \sgmgT;	\smile \sgmscap;	\Vdash \sgmVdash;
\cup \sgmcup;	\gg \sgmLap;	\smile \sgmsccue;	\Vdash \sgmvDash;
\dashv \sgmdashv;	\gg \sgmldot;	\smile \sgmsce;	\vee \sgmveebar;
\ecir \sgmecir;	\gg \sgmLE;	\smile \sgmscsim;	\triangleleft \sgmvltri;
\ecolon \sgmecolon;	\gg \sgmLEg;	\smile \sgmsfrown;	\propto \sgmvprop;
\eDot \sgmeDot;	\gg \sgmleg;	\smile \sgmsmid;	\triangleright \sgmvrtri;
\esdot \sgmesdot;	\gg \sgmlEg;	\smile \sgmsmile;	\Vdash \sgmVvdash;
\efDot \sgmefDot;	\gg \sgmlEg;	\smile \sgmspar;	
\megs \sgmegs;	\gg \sgmlg;	\sqsubset \sgmsqsub;	
\mels \sgmels;	\ll \sgmLl;	\sqsubseteq \sgmsqsube;	

7.7 iso-grk1.def

The greek macros are only used if you specify `\havegreek` in your document. I added this protection to guard against loading even more fonts into T_EX that may not be needed.

Also, the entities from greek 1 and greek 2 are intended for words, not symbols, which is why the macros typeset the lower cased ones in roman type.

These try to follow the style of surrounding text, both in shape and in size (same applies to the cyrillic entities that will follow).

α \sgmagr;	H \sgmEEgr;	ξ \sgmxgr;	υ \sgmugr;
A \sgmAgr;	θ \sgmthgr;	Ξ \sgmXgr;	Υ \sgmUgr;
β \sgmbgr;	Θ \sgmTHgr;	\omicron \sgmogr;	φ \sgmphgr;
B \sgmBgr;	ι \sgmiigr;	O \sgmOgr;	Φ \sgmPHgr;
γ \sgmggr;	I \sgmIgr;	π \sgmpgr;	χ \sgmkhgr;
Γ \sgmGgr;	κ \sgmkgr;	Π \sgmPgr;	X \sgmKHgr;
δ \sgmdgr;	K \sgmKgr;	ρ \sgmrgr;	ψ \sgmpsgr;
Δ \sgmDgr;	λ \sgmlgr;	P \sgmRgr;	Ψ \sgmPSgr;
ϵ \sgmegr;	Λ \sgmLgr;	σ \sgmsgr;	ω \sgmohgr;
E \sgmEgr;	μ \sgmmgr;	Σ \sgmSgr;	Ω \sgmOHgr;
ζ \sgmzgr;	M \sgmMgr;	ς \sgmsfgr;	
Z \sgmZgr;	ν \sgmngr;	τ \sgmtgr;	
η \sgmeegr;	N \sgmNgr;	T \sgmTgr;	

7.8 iso-grk2.def

$\acute{\alpha}$ \sgmaacr;	$\grave{\iota}$ \sgmidigr;	O \sgmOacr;	$\acute{\omega}$ \sgmohacr;
'A \sgmAacr;	$\grave{\text{I}}$ \sgmIdigr;	$\ddot{\text{U}}$ \sgmudigr;	' Ω \sgmOHacr;
$\acute{\epsilon}$ \sgmeacr;	$\grave{\text{i}}$ \sgmiacr;	$\ddot{\text{Y}}$ \sgmUdigr;	
'E \sgmEacr;	'I \sgmIacr;	$\acute{\upsilon}$ \sgmuacr;	
$\grave{\eta}$ \sgmeeacr;	$\grave{\text{i}}$ \sgmidiagr;	'Y \sgmUacr;	
'H \sgmEEacr;	$\acute{\omicron}$ \sgmoacr;	$\ddot{\text{U}}$ \sgmudiagr;	

7.9 iso-grk3.def

α \sgmalpha;	η \sgmeta;	ξ \sgmxi;	υ \sgmupsi;
β \sgmbeta;	θ \sgmthetas;	Ξ \sgmXi;	Υ \sgmUpsi;
γ \sgmgamma;	Θ \sgmTheta;	π \sgmpi;	ϕ \sgmphis;
Γ \sgmGamma;	ϑ \sgmthetav;	ϖ \sgmpiv;	Φ \sgmPhi;
F \sgmgammad;	ι \sgmiota;	Π \sgmPi;	φ \sgmphiv;
δ \sgmdelta;	κ \sgmkappa;	ρ \sgmrho;	χ \sgmchi;
Δ \sgmDelta;	κ \sgmkappav;	ϱ \sgmrhov;	ψ \sgmpsi;
ϵ \sgmepsi;	λ \sgmlambda;	σ \sgmsigma;	Ψ \sgmPsi;
ε \sgmepsiv;	Λ \sgmLambda;	Σ \sgmSigma;	ω \sgmomega;
ϵ \sgmepsis;	μ \sgmmu;	ς \sgmsigmav;	Ω \sgmOmega;
ζ \sgmzeta;	ν \sgmnu;	τ \sgmtau;	

7.10 iso-grk4.def

Wat is a variation? The common interpretation seems to be boldface, so these are all boldface (hence the 'b' in the name).

α \sgmbdalpha;	η \sgmbdeta;	ξ \sgmbdxi;	υ \sgmbdupsi;
β \sgmbdbeta;	θ \sgmbdthetas;	Ξ \sgmbdXi;	Υ \sgmbdUpsi;
γ \sgmbdgamma;	Θ \sgmbdTheta;	π \sgmbdpi;	ϕ \sgmbdphis;
Γ \sgmbdGamma;	ϑ \sgmbdthetav;	ϖ \sgmbdpiv;	Φ \sgmbdPhi;
F \sgmbdgammad;	ι \sgmbdiota;	Π \sgmbdPi;	φ \sgmbdphiv;
δ \sgmbddelta;	κ \sgmbdkappa;	ρ \sgmbdrho;	χ \sgmbdchi;
Δ \sgmbdDelta;	κ \sgmbdkappav;	ϱ \sgmbdrhov;	ψ \sgmbdpsi;
ϵ \sgmbdepsi;	λ \sgmbdlambda;	σ \sgmbdsigma;	Ψ \sgmbdPsi;
ε \sgmbdepsiv;	Λ \sgmbdLambda;	Σ \sgmbdSigma;	ω \sgmbdomega;
ϵ \sgmbdepsis;	μ \sgmbdmu;	ς \sgmbdsigmav;	Ω \sgmbdOmega;
ζ \sgmbdzeta;	ν \sgmbdnu;	τ \sgmbdtau;	

7.11 iso-lat1.def

á \sgmaacute;	ð \sgmeth;	ï \sgmiuml;	ß \sgmszlig;
Á \sgmAacute;	Ð \sgmETH;	Ï \sgmIuml;	þ \sgmthorn;
â \sgmacirc;	é \sgmeacute;	Ñ \sgmntilde;	Þ \sgmTHORN;
Â \sgmAcirc;	É \sgmEacute;	Ñ̄ \sgmNtilde;	ú \sgmuacute;
à \sgmagrave;	ê \sgmecirc;	ó \sgmoacute;	Ú \sgmUacute;
À \sgmAgrave;	Ê \sgmEcirc;	Ó \sgmOacute;	û \sgmucirc;
å \sgmaring;	è \sgmegrave;	ô \sgmocirc;	Û \sgmUcirc;
Å \sgmAring;	È \sgmEgrave;	Ô \sgmOcirc;	ù \sgmugrave;
ã \sgmatilde;	ë \sgmeuml;	ò \sgmograve;	Û̄ \sgmUgrave;
Ã \sgmAtilde;	Ë \sgmEuml;	Ò \sgmOgrave;	ü \sgmuuml;
ä \sgmauml;	í \sgmiacute;	ø \sgmoslash;	Ü \sgmUuml;
Ä \sgmAuml;	Í \sgmIacute;	Ø \sgmOslash;	ý \sgmyacute;
æ \sgmaelig;	î \sgmicirc;	ō \sgmotilde;	Ý \sgmYacute;
Æ \sgmAElig;	Î \sgmIcirc;	Õ \sgmOtilde;	ÿ \sgmyuml;
ç \sgmccedil;	ì \sgmigrave;	ö \sgmouml;	
Ç \sgmCcedil;	Ì \sgmIgrave;	Ö \sgmOuml;	

7.12 iso-lat2.def

The \sgmdcaron; (d') is correct but ugly.

\sgmjcirc; (j), \sgmeng; (m) and \sgmENG; (M) fail with common PostScript fonts. I decided to ignore the error for the moment since they are very rarely used. Borrowing the eng's from CM looked even worse than the black box :-)

ă \sgmabreve;	Ĝ \sgmGcirc;	ń \sgmnacute;	† \sgmtcedil;
Ă \sgmAbreve;	ĝ \sgmgdot;	Ń \sgmNacute;	‡ \sgmTcedil;
ā \sgmamacr;	Ĝ̄ \sgmGdot;	■ \sgmeng;	‡ \sgmTstrok;
Ā \sgmAmacr;	ĥ \sgmhcirc;	■ \sgmENG;	ƒ \sgmTstrok;
ą \sgmaogon;	Ĥ \sgmHcirc;	'n \sgmnapos;	Û \sgmUbreve;
Ą \sgmAogon;	h \sgmhstrok;	ñ \sgmncaron;	Ũ \sgmUbreve;
ć \sgmcacute;	H \sgmHstrok;	Ñ \sgmNcaron;	ů \sgmudblac;
Ć \sgmCacute;	ı̇ \sgmIdot;	ņ \sgmncedil;	Ů \sgmUdblac;
č \sgmccaron;	Ī \sgmImacr;	Ň \sgmNcedil;	ū \sgmumacr;
Č \sgmCcaron;	ī \sgmimacr;	ô \sgmodblac;	Ū \sgmUmacr;
ĉ \sgmccirc;	ij̇ \sgmijlig;	Õ \sgmOdblac;	u \sgmuogon;
Ĉ \sgmCcirc;	Ij̇ \sgmIJlig;	Ō \sgmOmacr;	U̇ \sgmUogon;
ċ \sgmcdot;	ı̇ \sgmiogon;	ō \sgmomacr;	û \sgmuring;
Č̇ \sgmCdot;	ı̇ \sgmIogon;	œ \sgmoelig;	Ů̇ \sgmUring;
d' \sgmdcaron;	ĩ \sgmitilde;	Œ \sgmOElig;	ü \sgmutilde;
Ď \sgmDcaron;	Ī̇ \sgmItilde;	ř \sgmracute;	Ů̇ \sgmUtilde;
ď \sgmdstrok;	ĵ \sgmjcirc;	Ř \sgmRacute;	ŵ \sgmwcirc;
Đ \sgmDstrok;	Ĵ \sgmJcirc;	ř̇ \sgmrcaron;	Ŵ \sgmWcirc;
ě \sgmecaron;	ķ \sgmkcedil;	Ř̇ \sgmRcaron;	ÿ \sgmycirc;
Ě \sgmEcaron;	Ķ \sgmKcedil;	ı̇ \sgmrcedil;	Ŷ \sgmYcirc;
é \sgmedot;	Ķ̇ \sgmkgreen;	Ŕ \sgmRcedil;	ÿ̇ \sgmYuml;
Ě̇ \sgmEdot;	Í \sgmlacute;	ś \sgmsacute;	ż \sgmzacute;
ē \sgmemacr;	Ĺ \sgmLacute;	Ś \sgmSacute;	Ż \sgmZacute;
Ē \sgmEmacr;	ł̇ \sgmlcaron;	š \sgmscaron;	ž \sgmzcaron;
ę \sgmeogon;	Ł̇ \sgmLcaron;	Š \sgmScaron;	Ž \sgmZcaron;
Ę \sgmEogon;	ł̇ \sgmlcedil;	ş \sgmscedil;	z \sgmzdot;
ğ \sgmgacute;	Ł̇ \sgmLcedil;	Ş \sgmScedil;	Ž̇ \sgmZdot;
ĝ \sgmgbreve;	ł̇ \sgmlmidot;	ṧ \sgmscirc;	
Ğ \sgmGbreve;	Ł̇ \sgmLmidot;	Ŝ̇ \sgmScirc;	
ġ \sgmgcedil;	ł̇ \sgmlstrok;	ť \sgmtcaron;	
ġ̇ \sgmgcirc;	Ł̇ \sgmLstrok;	Ť \sgmTcaron;	

7.13 iso-dia.def

Used an 'a' to show the composition, but this is not the way they are typically used in SGML, so maybe these should be changed to have their natural width (they are now 'real' accents)

á \sgmacute;a	â \sgmcirc;a	à \sgmgrave;a	ã \sgmtilde;a
ä \sgmbreve;a	ã \sgmdblac;a	ā \sgmmacr;a	ä \sgmuuml;a
ǎ \sgmcaron;a	ä \sgmdie;a	ą \sgmogon;a	
ç \sgmcedil;a	à \sgmdot;a	å \sgmring;a	

7.14 iso-pub.def

The grey boxes depend on PostScript output, and of course there is no ‘fj’ ligature in most fonts. The \sgmrx; (R) is very wrong indeed, but it hard to align a composite across different font families. Don’t know how to solve that yet (looks OK in CM though).

\sgmemsp;	$\frac{5}{6}$ \sgmlfracfivesix;	♡ \sgmhearts;	... \sgmmlldr;
\sgmensp;	$\frac{c}{o}$ \sgmincare;	♠ \sgmspades;	“ \sgmrdquor;
\sgmemsponethree;	■ \sgmblock;	✦ \sgmmalt;	‘ \sgmrsquor;
\sgmemsponefour;	▪ \sgmuhblk;	† \sgmdagger;	∴ \sgmvellip;
\sgmnumsp;	■ \sgmlhblk;	‡ \sgmDagger;	– \sgmhybull;
\sgmpuncsp;	▒ \sgmblkonefour;	✓ \sgmcheck;	◇ \sgmloz;
\sgmthinsp;	▒ \sgmblkonetwo;	✗ \sgmxcross;	◆ \sgmlozf;
\sgmhairsp;	■ \sgmblkthreefour;	‡ \sgmsharp;	◁ \sgmltri;
— \sgmmdash;	■ \sgmmarker;	b \sgmflat;	▷ \sgmrtri;
- \sgmndash;	○ \sgmcir;	♂ \sgmmale;	★ \sgmstarf;
- \sgmdash;	□ \sgmsqu;	♀ \sgmfemale;	♠ \sgmnatur;
□ \sgmblank;	□ \sgmrect;	☎ \sgmphone;	℞ \sgmrx;
... \sgmhellip;	△ \sgmutri;	☎ \sgmtelrec;	* \sgmsex;
∴ \sgmldr;	▽ \sgmdtri;	Ⓟ \sgmcopy;	⊕ \sgmtarget;
\sgmlfraconefour;	☆ \sgmstar;	◁ \sgmcaret;	— \sgmdlcrop;
\sgmlfraconefive;	● \sgmbull;	, \sgmlsquor;	\sgmdrcrop;
\sgmlfraconehalf;	■ \sgmsquf;	„ \sgmldquor;	\sgmdrcrop;
\sgmlfraconefour;	▲ \sgmutrif;	ff \sgmfflig;	\sgmulcrop;
\sgmlfraconehalf;	▼ \sgmdtrif;	fi \sgmfilig;	\sgmulcrop;
\sgmlfraconefour;	◀ \sgmltrif;	fj \sgmfjlig;	\sgmurcrop;
\sgmlfraconefour;	▶ \sgmrtrif;	ffi \sgmffilig;	
\sgmlfraconefour;	♣ \sgmclubs;	ffi \sgmffilig;	
\sgmlfraconefour;	◇ \sgmdiams;	fi \sgmffilig;	

7.15 iso-num.def

\sgmlowbar; () is actually an underlined emspace.

An option should be added to use fraction fonts if available.

1 \sgmfraconefour;	⊞ \sgmcurrent;	Ω \sgmohm;	¡ \sgmiexcl;
2 \sgmfraconefour;	£ \sgmpound;	° \sgmdeg;	’ \sgmquot;
3 \sgmfraconefour;	\$ \sgmdollar;	◊ \sgmordm;	’ \sgmapos;
4 \sgmfraconefour;	¢ \sgmcent;	ª \sgmordf;	(\sgmlpar;
5 \sgmfraconefour;	¥ \sgmyen;	§ \sgmsect;) \sgmrpar;
6 \sgmfraconefour;	# \sgmnum;	¶ \sgmpara;	, \sgmcomma;
7 \sgmfraconefour;	% \sgmpercnt;	· \sgmmiddot;	— \sgmlowbar;
8 \sgmfraconefour;	& \sgmamp;	← \sgmlarr;	- \sgmhyphen;
9 \sgmfraconefour;	* \sgmast;	→ \sgmrarr;	. \sgmperiod;
1 \sgmlsupone;	@ \sgmcommat;	↑ \sgmuarr;	/ \sgmsol;
2 \sgmlsuptwo;	[\sgmlsqb;	↓ \sgmdarr;	: \sgmcolon;
3 \sgmlsupthree;	\ \sgmbsol;	© \sgmcopy;	; \sgmsemi;
+ \sgmplus;] \sgmrsqb;	® \sgmreg;	? \sgmquest;
± \sgmplusmn;	{ \sgmlcub;	™ \sgmtrade;	¿ \sgmiquest;
< \sgmlt;	— \sgmhorbar;	¡ \sgmbrvbar;	« \sgmlaquo;
= \sgmequals;	\sgmverbar;	¬ \sgmnot;	» \sgmraquo;
> \sgmgt;	} \sgmrcub;	↓ \sgmsung;	‘ \sgmlsquo;
÷ \sgmdivide;	μ \sgmmicro;	! \sgmexcl;	’ \sgmrsquo;
× \sgmtimes;			

“ \sgmldquo; \sgmnbsp;
 ” \sgmrdquo; \sgmshy;

7.16 iso-tech.def

ℵ	\sgmaleph;	↔	\sgmiff;	⊥	\sgmperp;	Å	\sgmangst;
^	\sgmand;	∞	\sgminfin;	'	\sgmprime;	ℬ	\sgmbernou;
⊥	\sgmangninezero;	∫	\sgmint;	"	\sgmPrime;	◦	\sgmcompfn;
⋈	\sgmangsph;	∈	\sgmisin;	α	\sgmprop;	⋯	\sgmDot;
≈	\sgmap;	⟨	\sgmlang;	√	\sgmradic;	⋯	\sgmDotDot;
∴	\sgmbecaus;	←	\sgmlArr;)	\sgmrang;	ℋ	\sgmhamilt;
⊥	\sgmbottom;	≤	\sgmle;	⇒	\sgmrArr;	ℒ	\sgmlagran;
∩	\sgmcap;	−	\sgmminus;	~	\sgmsim;	*	\sgmlowast;
≅	\sgmcong;	⊕	\sgmnplus;	≈	\sgmsime;	∉	\sgmnotin;
ϕ	\sgmconint;	∇	\sgmnabla;	□	\sgmsquare;	o	\sgmorder;
∪	\sgmcup;	≠	\sgmne;	⊂	\sgmsub;	ℳ	\sgmphmmat;
≡	\sgmequiv;	∃	\sgmni;	⊆	\sgmsube;	⋯	\sgmtdot;
∃	\sgmexist;	∨	\sgmor;	⊃	\sgmsup;	'''	\sgmtrprime;
∀	\sgmforall;	∥	\sgmpar;	⊇	\sgmsupe;	≜	\sgmwedgeq;
f	\sgmfnof;	∂	\sgmpart;	∴	\sgmtherefour;		
≥	\sgmge;	‰	\sgmpermil;	∥	\sgmVerbar;		

7.17 iso-box.def

Don't use these to draw boxes! I only wrote these to be complete, and frankly i don't see a real application for these blocks in print, as drawing is font-dependent in every possible way.

—	\sgmboxh;	┌	\sgmboxvR;	⌋	\sgmboxVR;	⌌	\sgmboxUL;
	\sgmboxv;	└	\sgmboxhU;	⌋	\sgmboxHU;	⌍	\sgmboxdL;
┌	\sgmboxur;	┐	\sgmboxvL;	⌋	\sgmboxVL;	⌎	\sgmboxDr;
└	\sgmboxul;	┌	\sgmboxhD;	⌋	\sgmboxHD;	⌏	\sgmboxUr;
┐	\sgmboxdl;	└	\sgmboxvH;	⌋	\sgmboxVH;	⌐	\sgmboxuL;
┌	\sgmboxdr;	≡	\sgmboxH;	⌋	\sgmboxVr;	⌑	\sgmboxDl;
└	\sgmboxvr;	∥	\sgmboxV;	⌋	\sgmboxHu;	⌒	\sgmboxdR;
└	\sgmboxhu;	⌌	\sgmboxUR;	⌋	\sgmboxVl;		
┐	\sgmboxvl;	⌍	\sgmboxUL;	⌋	\sgmboxHd;		
┌	\sgmboxhd;	⌎	\sgmboxDL;	⌋	\sgmboxVh;		
└	\sgmboxvh;	⌏	\sgmboxDR;	⌋	\sgmboxuR;		

7.18 iso-cyr1.def

The cyrillic fonts will only be loaded after you give the command \havecyrillic.

a	\sgmacy;	Ё	\sgmIOcy;	м	\sgmmcy;	Т	\sgmTcy;
А	\sgmAcy;	Ж	\sgmzhcy;	М	\sgmMcy;	У	\sgmUcy;
б	\sgmbcy;	Ѡ	\sgmZHcy;	н	\sgmnicy;	Ф	\sgmficy;
Б	\sgmBcy;	з	\sgmzcy;	Н	\sgmNicy;	Φ	\sgmFicy;
в	\sgmvicy;	З	\sgmZcy;	о	\sgmoccy;	х	\sgmkhicy;
В	\sgmVicy;	и	\sgmiccy;	О	\sgmOcy;	Х	\sgmKHicy;
г	\sgmgicy;	И	\sgmIicy;	п	\sgmpicy;	ц	\sgmtscy;
Г	\sgmGicy;	Й	\sgmjicy;	П	\sgmPicy;	Ц	\sgmTScy;
д	\sgmdicy;	Ў	\sgmJicy;	р	\sgmricy;	ч	\sgmchicy;
Д	\sgmDicy;	к	\sgmkicy;	Р	\sgmRicy;	Ч	\sgmCHicy;
e	\sgmieicy;	К	\sgmKicy;	с	\sgmsicy;	ш	\sgmshicy;
Е	\sgmIEicy;	л	\sgmlicy;	С	\sgmScy;	Ш	\sgmSHicy;
ё	\sgmiocy;	Л	\sgmLicy;	т	\sgmticy;		

щ \sgmshchcy;	ы \sgmycy;	э \sgmecy;	я \sgmyacy;
Ш \sgmSHCHcy;	Ы \sgmYcy;	Э \sgmEcy;	Я \sgmYAcy;
ъ \sgmhardcy;	ь \sgmsoftcy;	ю \sgmyucy;	№ \sgmnumero;
Ъ \sgmHARDcy;	Ь \sgmSOFTcy;	Ю \sgmYUcy;	

7.19 iso-cyr2.def

ђ \sgmdjcy;	Ѕ \sgmDScy;	љ \sgmljcy;	Ќ \sgmKJcy;
Ђ \sgmDJcy;	Ї \sgmiukcy;	Љ \sgmLJcy;	Ў \sgmubrky;
ѓ \sgmgjcy;	І \sgmIukcy;	њ \sgmnjcy;	Ў \sgmUbrcy;
Ѓ \sgmGJcy;	Њ \sgmyicy;	Њ \sgmNJcy;	џ \sgmdzcy;
е \sgmjukcy;	Ќ \sgmYIcy;	ћ \sgmtshcy;	Ѣ \sgmDZcy;
Є \sgmJukcy;	Ј \sgmjsercy;	Ћ \sgmTSHcy;	
ѕ \sgmdscy;	Ј \sgmJsercy;	ќ \sgmkjcy;	

The \LaTeX Graphics Companion

Illustrating documents with \TeX and PostScript

Michel Goossens

CERN, Geneva, Switzerland

Sebastian Rahtz

Elsevier Science Ltd, Oxford, United Kingdom

Frank Mittelbach

\LaTeX 3 Project, Mainz, Germany

Abstract

This handy reference describes techniques and tricks needed to illustrate \LaTeX documents, and answers common user questions about graphics and PostScript fonts. It provides the first full description of the standard \LaTeX color and graphics packages, and shows how you can combine \TeX and PostScript capabilities to produce beautifully illustrated pages. Following the successful format of *The \LaTeX Companion*, this new book will be an invaluable \LaTeX resource for incorporating pictures into text.

Keywords: Abstract, Table of Contents, Preface, for The \LaTeX Graphics Companion

You will learn how to: Incorporate graphic files into a \LaTeX document

- Program technical diagrams using several different languages
- Produce color pictures
- Achieve special effects with fragments of embedded PostScript
- Make high-quality musical scores and games diagrams

You will find detailed descriptions of: Important packages like XYPIC, PSTricks, and METAPOST

- The standard LaTeX color and graphics packages
- PostScript fonts and how to use them in \LaTeX
- The dvips dvi to PostScript driver
- Ghostscript, the free interpreter that lets you view or print PostScript files, even if you do not have a PostScript printer

The authors examine a number of packages that extend or modify \LaTeX 's basic illustration features, and present hundreds of examples of useful solutions to graphics and font problems. In addition to pack-

ages for general drawing, the book also presents specific tools for mathematicians, physicists, chemists, engineers, and for people interested in games and music typesetting.

All the packages and programs described in this book are freely available in public software archives, and the source code for all examples has been placed on CTAN, the \TeX archives. (Details in Appendix B)

Michel Goossens is the President of the TeX Users Group (TUG) and a coauthor of *The \LaTeX Companion*. His work responsibilities have included LaTeX support for other scientists. Sebastian Rahtz is Secretary of TUG, a cofounder of the CTAN archives, and creator of the \TeX Live CD-ROM. Frank Mittelbach is Manager of the \LaTeX 3 Project, in which capacity he oversaw the release of \LaTeX 2 ϵ . He, too, is a coauthor of *The LaTeX Companion*, as well as the editor of a series of publications on *Tools and Techniques for Computer Typesetting*.

For more information about this and other books: <http://www.awl.com/cseng>

Table of Contents

Preface	xxi	Spanning braces	49
1 Graphics with \LaTeX	1	2.4 Other packages in the graphics bundle	50
1.1 Graphics systems and typesetting	2	2.4.1 The epsfig and rotating packages	50
1.2 Drawing types	3	2.4.2 The lscape package	51
1.3 \TeX 's interfaces	6	2.4.3 The trig package	51
1.3.1 Methods of integration	6	2.4.4 The keyval package	51
Using special commands	7	3 Working with METAFONT and METAPOST	53
Using fonts	7	3.1 The META language	54
Using half-tones	9	3.1.1 Examples of META programs	55
1.3.2 Methods of manipulation	9	3.2 Using the META programs	60
1.3.3 \TeX 's graphic hooks	9	3.2.1 Running METAFONT	60
Using \TeX 's built-in commands	9	3.2.2 An alternative to METAFONT —	
Using fonts	10	METAPOST	62
Using special commands	10	Adding text	64
1.4 Graphics languages	11	Adding text — some gory details	66
1.4.1 \TeX -based graphics languages	11	3.3 METAPOST macro libraries	68
Character-based diagrams and pictures	12	3.3.1 Boxing macros	68
$\text{Pic}\TeX$	14	3.3.2 The METAPOST graph package	72
The dratex and aldratex packages	17	Frames, ticks, grids, and scales	75
Building on the picture environment	17	Reading data files	76
XYpic	19	Different graph types	78
PSTricks	19	3.4 \LaTeX interfaces to META	83
1.4.2 External graphics languages and drawing programs	20	3.4.1 The mfpic package	84
HPGL	20	Basic drawing commands	86
gnuplot — a plotting package	20	Drawing parameters	87
The pic language	22	Global modifier commands	88
Special-purpose external graphics languages	23	Figure modifier commands	88
METAFONT and METAPOST	25	Analytical curves and functions	89
1.5 Choosing a package	25	Annotating drawings	89
2 The $\LaTeX 2_{\epsilon}$ graphics bundle	27	Examples	90
2.1 Loading the graphics packages	27	Going further	92
2.1.1 Package device driver options	28	3.5 mftoeps: a direct link between META FONT and PostScript	93
2.2 Inclusion of graphics files	29	4 Harnessing PostScript inside \LaTeX : the PSTricks package	95
2.2.1 <code>includegraphics</code> syntax in the graphics package	30	4.1 The components of PSTricks	95
2.2.2 <code>includegraphics</code> syntax in the <code>graphicx</code> package	32	4.2 Basic PSTricks concepts	97
2.2.3 Setting default key values for the <code>graphicx</code> package	38	4.2.1 Commands and arguments	97
2.2.4 Declarations guiding the inclusion of images	39	4.2.2 Setting graphics parameters	98
2.3 Graphical manipulation of \LaTeX objects	41	4.2.3 Coordinates and units	98
2.3.1 Scaling a \LaTeX box	42	4.2.4 Coloring objects	100
2.3.2 Resizing to a given size	43	4.3 The graphic objects	100
2.3.3 Rotating a \LaTeX box	44	4.4 Examples of basic graphic objects	102
2.3.4 Combining effects	48	4.5 Mixing text and graphics	107
2.3.5 Applications	48	4.6 Nodes and their connections, and trees	114
Rotating tabular material	48	4.6.1 Matrices — grid-based nodes	121
		4.6.2 Tree diagrams	125
		4.6.3 Specifying tree bounding boxes	132

4.7	Data plotting	132	6	Applications in chemistry, physics, and engineering	205
4.8	Working with a third dimension	135			
4.8.1	Iterating commands	139	6.1	Typographical rules for scientific texts	205
4.9	Customizing and programming PSTricks	140	6.1.1	Typesetting chemical symbols	207
4.9.1	New PSTricks objects and styles	140	6.2	The XyMTeX system	208
4.9.2	PSTricks programming examples	145	6.2.1	General conventions	210
4.10	Other PSTricks tools	153		Syntax for specific commands	210
4.11	Driver configuration for PSTricks	153		Syntax for general commands	212
4.12	Summary of PSTricks commands and parameters	154		Charges on atoms	213
4.12.1	PSTricks basic drawing commands	154	6.2.2	The carom package	214
4.12.2	PSTricks node-drawing commands	159	6.2.3	The lowcycle package	215
	Node creators	159	6.2.4	The hetarom and hetaromh packages	216
	Node connectors	160	6.2.5	The ccycle and hcycle packages	218
	Coil and zigzag node connectors	160	6.2.6	The aliphath package	219
4.12.3	PSTricks node connection labeling commands	161	6.2.7	Combining structures	220
	Labeling based on connector length	161	6.2.8	Defining your own structures	220
	Labeling based on distance between nodes	161	6.3	The ppchtex package	222
	Labeling nodes	161	6.3.1	Structures	222
4.12.4	PSTricks drawing commands comparable to node connectors	163	6.3.2	Definitions	224
4.12.5	PSTricks tree-drawing commands	163	6.3.3	Bonds	225
4.12.6	PSTricks plotting commands	165	6.3.4	Combinations	225
	Axes	165	6.3.5	Chemical equations	227
4.12.7	PSTricks 3D commands	166	6.3.6	Special features	227
5	The XYPIC package	167	6.4	Drawing Feynman diagrams	228
5.1	Introducing XYPIC	167	6.4.1	Using FeynMF	229
5.2	A first example of XYPIC code	168	6.4.2	Writing FeynMF diagrams	231
5.3	Basic constructs	169		Vertex mode and algorithmic layout	231
5.3.1	Initial positions	169		Immediate mode	237
5.3.2	Making connections	170	6.4.3	Extending FeynMF	239
5.3.3	Dropping objects	171	6.5	Typesetting timing diagrams	239
5.3.4	Entering text in your pictures	173	6.5.1	Commands in the timing environment	239
5.4	Extensions	174	6.5.2	Customization	243
5.4.1	Curves and splines	175	6.6	Electronics and optics diagrams	243
5.4.2	Frames and brackets	176	6.6.1	General circuit diagram commands	245
5.5	Features	178	6.6.2	Examples	247
5.5.1	Arrows	178	6.7	Using the m4 macro processor for electronics diagrams	250
5.5.2	Matrix-like diagrams	180	6.7.1	Basic principles	250
5.5.3	Graphs	184	6.7.2	Customizing the diagram	251
5.5.4	Two-cell diagrams	188	7	Preparing music scores	253
5.5.5	Polygons	190	7.1	Using \TeX for scores—an overview	254
5.5.6	Arcs, circles, and ellipses	195	7.2	Using MusiX \TeX	255
	Simple circles and ellipses	195	7.2.1	The structure of a MusiX \TeX source	255
	Constructing arcs	196	7.2.2	Writing notes	256
5.5.7	Lattices and web structures	197		Specifying note pitches	258
5.5.8	Links and knots	198		Note symbols	258
	Constructing crossings	199	7.2.3	Note spacing	260
	Adding joins	200	7.2.4	A moderately complete example	261
			7.2.5	Running Musix \TeX	263
			7.3	abc2mtex—easy writing of tunes	264
			7.3.1	Writing abc2mtex source	265
			7.3.2	Running abc2mtex	268

7.4	MPP, a Musix \TeX preprocessor	269	9.3.1	The colortbl package	326
7.4.1	Writing MPP-source	269	9.3.2	Examples	329
7.4.2	Running MPP	273	9.4	Color slides with \LaTeX : the seminar class	338
7.5	miditex	274	9.4.1	Using the seminar class	339
7.5.1	Running miditex	274	9.4.2	Frame styles	339
8	Playing games	277	9.4.3	Interleaving notes and selecting subsets	341
8.1	Chess	277	9.4.4	Controlling slide size, fonts, and magnification	341
8.1.1	A METAFONT chess font	278	9.4.5	Fonts	342
8.1.2	chess — a package for typesetting chess	278	9.4.6	The local control file	342
	Annotating games	279	9.5	Color in the printing industry and separation	346
	Tournament style	280	9.5.1	Color separation	347
8.1.3	Extensions to the chess package	284	9.5.2	Color separation using \LaTeX and dvips	348
8.1.4	Interfacing chess databases	285	10	Using PostScript fonts	351
8.1.5	Using Adobe's cheq font	288	10.1	Using preconfigured PostScript fonts	353
8.2	Xiangqi — Chinese chess	292	10.1.1	The PSNFSS system	353
8.3	Go	295		Documents containing math	354
8.3.1	Possible problems	298		Symbol font support	356
8.4	Backgammon	299	10.1.2	Fonts, metric files — the whole lot	359
8.5	Card games	302	10.1.3	Installing PostScript fonts	362
8.6	Bridge	303	10.2	\TeX font technology	363
8.6.1	Card deals	303	10.2.1	Types of \TeX fonts	363
8.6.2	Bidding	305		The gf font format	363
8.7	Crosswords	306		The pxl font format	364
8.7.1	Recent developments	309		The pk font format	364
9	The world of color	311	10.2.2	\TeX font metric files	364
9.1	An introduction to color	312	10.2.3	\TeX virtual fonts	367
9.1.1	Color theories	312	10.3	PostScript font technology	370
9.1.2	Color systems	313	10.3.1	Types of PostScript fonts	370
	The additive RGB color space	313		Type 1 fonts	370
	The subtractive CMYK color space	314		Composite fonts	372
9.1.3	Symbolic values of color	314		Multiple Master fonts	372
9.1.4	Color harmonies	314		Type 3 fonts	374
	Two-color harmonic combinations	316		TrueType fonts	375
	Three-color harmonic combinations	316	10.3.2	Adobe Font Metric files	375
	Four-color harmonic combinations	316	10.3.3	Encoding of PostScript fonts	378
9.1.5	Color and readability	316	10.3.4	Rendering PostScript fonts	379
9.2	Colors and \LaTeX — the color package	317		Turning outlines into pixels	379
9.2.1	Supported options	317		The Adobe Type Manager	380
	The pstcol package	318	10.4	Classifying PostScript fonts	381
9.2.2	Using colors	318	10.5	Setting up new PostScript fonts	386
	Using predefined colors	318	10.5.1	\TeX and encoding	386
	Specifying values for color models	319	10.5.2	The TeXBase1 encoding	386
	Defining colors	319	10.5.3	Making \TeX metrics for PostScript fonts	387
	Using named colors	320	10.5.4	AFM to TFM, and VF, conversion tools	387
	Special concerns with color in \LaTeX	320	10.5.5	The Y&Y font manipulation tools	389
9.2.3	Page color	321	10.5.6	The afm2tfm program	390
9.2.4	Colored box backgrounds	322	10.5.7	The fontinst package	393
9.2.5	Calculating colors	324			
9.3	Coloring tables	326			

The latinfamily interface	394	11.4.1 Ghostscript options and initialization	443
The installfont interface	394	11.4.2 Ghostscript and fonts	446
Metric files	396	11.4.3 Using different devices with Ghostscript	447
Encoding files	398	11.4.4 Interactive Ghostscript versions	450
Examples of installfont command files	399	11.4.5 Ghostscript applications	450
10.6 Multiple Master fonts—a case study	406	Extracting text from PostScript files	451
10.6.1 Generating instance files using Ghostscript	408	Creating EPS Interchange files	452
10.6.2 The Multiple Master setup of this book	410	Preparing Adobe Illustrator files	452
		Working with PDF (Portable Document Format)	453
11 PostScript drivers and tools	413	11.5 PostScript font to PK font format conversion	453
11.1 Introduction to DVI drivers	413	11.5.1 The ps2pk program	453
11.2 The dvips PostScript driver	415	11.5.2 The gsftopk program	455
11.2.1 Command line and configuration file options	415	11.6 Generating images for Web pages using dvips and Ghostscript	455
11.2.2 Paper sizes	422	11.7 PSfrag — adding labels to included pictures	460
11.2.3 special support	424		
Including PostScript files and literal PostScript code	425	A Technical appendixes	463
Color	427	A.1 emTeX “specialcommands”	463
Other special support	427	A.2 tpic “specialconventions”	464
11.2.4 Font support	428	A.3 The bm2font program	467
11.2.5 Special hooks	429	A.4 The dvips color separation header file	468
11.2.6 Debugging	432	A.5 Catalogue of typefaces with Fontname abbreviations	471
11.3 PostScript page-manipulation tools	433	A.6 Font encoding tables	474
11.3.1 The psutils suite	435	A.7 fontinst reference information	487
pstops : rearranging pages in a PostScript file	437	A.7.1 Encoding files	487
psnup : put multiple pages on a single sheet	438	A.7.2 Metric files	488
psbook : rearrange pages in a PostScript file into signatures	439	A.7.3 Low-level fontinst commands	491
psselect : selecting pages from a PostScript file	440	A.8 Ghostscript drivers	494
psmerge : merging PostScript files	440	B Getting all the goodies	497
psresize : scale and resize PostScript	440	B.1 Connecting to CTAN	498
Handling resource information in a PostScript file	441	B.1.1 Finding files on the archive	498
epsffit : fitting EPS files into a constrained size	442	B.2 Getting a package from the archive	498
11.4 Ghostscript, a PostScript interpreter	442	B.3 List of packages and programs	504

PREFACE

In this book we describe techniques and tricks of extended \LaTeX typesetting in the area of graphics and fonts. We examine how to draw pictures with \LaTeX and how to incorporate graphics files in a \LaTeX document. We explain how to program pictures using METAFONT and METAPOST as well as how to achieve special effects with small fragments of embedded PostScript. We show how to manipulate whole \TeX pages in PostScript and how to exploit all the features of PostScript Type 1 fonts. And, of course, we also look in detail at a whole range of tools for building graphics in \TeX itself.

\TeX is the world's premiere markup-based typesetting system, and PostScript is the leading language for describing the printed page. This book describes how they can produce even more beautiful results when they work together. \TeX 's mathematical capability, its paragraph building, its hyphenation and its programmable extensibility can cooperate with the graphical flexibility and the font-handling capabilities of PostScript, to provide a rich partnership for both author and typesetter. We describe dvi-to-PostScript drivers, look at tools for manipulating PostScript files, and discuss in detail the features of the free program Ghostscript which lets you view or print PostScript files even if you do not have a PostScript printer.

This volume is not a complete consumer guide to packages. In trying to teach by example we present hundreds of examples of the most useful types of solutions based on proven and well-known implementations. But, in the space available, we cannot provide a full manual for every package. Our main aim is to show how easy it is to use a given package and whether or not it seems to do what is required, not to dwell on precise details of syntax or run options. Nevertheless, we have described in more detail a few selected tools that we consider especially important.

We assume you know some \LaTeX ; you cannot read this book by itself if you have never used \TeX before. We recommend that you start with *\LaTeX : A Document Preparation System, Second Edition* (Lamport, 1994), and continue with *The \LaTeX Companion* to explore some of the many (non-graphical) packages available.

Why \LaTeX , and why PostScript?

This book is about \LaTeX , graphics, and PostScript. We believe that the structured approach of a system

like \LaTeX is the best way to use \TeX , and \LaTeX is by far the most widely used \TeX format. This means that it attracts those developing new packages (especially since the major revision of \LaTeX in 1993), and thus some of what we describe works only in \LaTeX (this applies especially to material to do with font selection schemes). We apologize in advance for our \LaTeX bias to those who appreciate the elegance of the original plain \TeX format and its derivatives and promise them that most of the packages work well with any \TeX dialect: the delights of systems like METAPOST, PSTricks, XYPIC, and Musix \TeX are open to all.

We also want to explain why we talk about PostScript so much. This language has been established for over a decade as an extremely flexible page-description language and it remains the tool of choice for professional typesetters. Among the features that make it so attractive are:

- the quantity, quality, and flexibility of Type 1 fonts;
- the device-independence and portability of files;
- the quality of graphics and the quantity of drawing packages that generate it;
- the facilities for manipulating text;
- the mature color-printing technology;
- the encapsulation conventions that make it easy to embed PostScript graphics;
- the availability of screen-based implementations (Display PostScript and Ghostscript/GhostView).

In the last few years PostScript has spawned an enterprising child, the PDF (Portable Document Format) language used by Adobe Acrobat. Designed for screen display with hypertext features, PDF offers a new degree of portability and efficiency. While we do not cover PDF here, \LaTeX works well in producing "rich" PDF documents, and special versions of \TeX that produce PDF directly will probably be available by the time this book is published.

Again, we apologize to those of you who are disappointed not to read about \LaTeX 's association with QuickDraw, TrueType fonts, the Windows GDI, HPGL, PCL, etc, but with so many packages available, we had to make a choice.

Please note that the absence of a given package or tool in this book in no way implies that we consider it as less useful or of inferior quality. We do think, though, that we have included a representative set of tools and packages, and we sincerely hope that you will find here one or more subjects to entertain you.

How this book is arranged

This book is subdivided in two basic ways: by application area, and by technique. We suggest that all readers look at Chapter 1 before going any further, since this introduces how we think about graphics and summarizes some techniques developed in later chapters. We also suggest you read Chapter 2, which covers the LaTeX standard graphics package, since the tools for including graphics files will be needed often.

We have tried to make it possible to read each of the other chapters separately; you may prefer to go straight to the chapters that cover your subject area or look at those that describe a particular tool. Three chapters are dedicated to particular programs and packages:

3. Working with METAFONT and METAPOST shows how to exploit the power of Knuth's METAFONT language; this can work well with LaTeX and provides unparalleled expressive power for describing many types of diagrams.
4. Harnessing PostScript inside \LaTeX : the PSTricks Package describes what we think is the most powerful system to integrate LaTeX closely with the PostScript language.
5. The XYPIC Package introduces a package that goes to great lengths to define a notation for many kinds of mathematics diagrams and implements it in a generic and portable way.

Then follow three chapters that discuss problems in special application areas and survey some packages in each:

6. Applications in chemistry, physics, and engineering looks at chemical bonds, Feynman diagrams, and circuit diagrams.
7. Preparing music scores with TeX considers the powerful MusixTeX package and some of the support programs that you can use to print beautiful musical scores.
8. Playing games is for those who use LaTeX for play as well as work; we show how to typeset chess boards, annotate diagrams for bridge or other card games, or even handle backgammon and Go.

Our last three chapters address areas of general interest, mostly not related to particular LaTeX packages or subject areas:

9. The world of color provides full details about the standard LaTeX color package and also describes other packages based on it. We also look at some issues of color design and problems related to color printing.
10. Using PostScript fonts describes the ins and outs of using PostScript fonts with LaTeX, including AFM files, virtual fonts, font naming, and Multiple Master fonts.
11. PostScript drivers and tools provides a complete reference to some of the most commonly used free software packages—dvips, ghostscript, the psutils suite, and programs to convert PostScript fonts to PK format.

Finally the appendixes contain technical information that complements the above chapters. They also give the Internet location of all software described in this book and show sample sessions to help you download it in different ways.

Using, and finding, all those packages and programs

We describe many packages and options that extend or modify \LaTeX 's basic possibilities. In order to show their action, we must in principle load them all at the same time. For various reasons, however, that is impractical, if not impossible. Indeed, many packages use a lot of internal variables for which most implementations of TeX have fixed limits. In producing this book we used a different strategy (pioneered with *The \LaTeX Companion*): we ran nearly all the examples as separate jobs and included them as EPS files. In almost all cases this was entirely automated—the code printed here is exactly that used to prepare the output. To help you run the examples yourself, each piece of \LaTeX code also shows the 'usepackage' commands needed to run it.

We have put the source of the examples on CTAN (Comprehensive TeX Archive Network—see Appendix B) in the directory info/lgc. Most examples are identified by a number enclosed in a box. This means that the exact code used to create the picture is available, with the identifying number as filename. Most files are in \LaTeX source format (with an extension of ltx) but there are also plain TeX files (extension ptx), METAPOST source files (extension mp), MusixTeX preprocessor source files (extensions abc and mpp), pic files (extension pic), and m4 sources (extension m4).

All the packages and programs described in this book are freely available in public software archives; some are in the public domain, while others are protected by copyright. Some programs are only available in source form or work only on certain computer platforms, and you should be prepared for a certain amount of “getting your hands dirty” in some cases. We also cannot guarantee that later versions of packages or programs will give results identical to those in our book. Many of them are under active development, and new or changed versions appear several times a year; we completed this book at the end of 1996, and tested the examples with the versions current at that time.

In Appendix B are given full details on how to access CTAN sites and how to download files using the Internet. You can also purchase a CD-ROM from the TeX Users Group containing implementations of TeX for various systems and many packages and fonts. This CD-ROM contains all the packages described in this book and *The LaTeX Companion*.

Acknowledgments

This book has two sources. First, it descends from a survey by Sebastian Rahtz of TeX and graphics originally prepared for the TeX eter conference in 1988, revised many times and distributed as a technical report in 1989 by Southampton University Computer Science Department, and from Rahtz’ ongoing support and documentation for using PostScript fonts in \LaTeX . Second, it descends from material that Michel Goossens, Frank Mittelbach, and Alexander Samarin could not include in *The LaTeX Companion*. It became clear, long before that book was finished, that it could not do full justice to graphics and PostScript and that another book (at least) would be needed. This book was conceived, born, and brought up at CERN, arising from text-processing support for this great laboratory by Michel Goossens, assisted in 1993–1994 by Sebastian Rahtz.

We have many people to thank. Our primary debt, of course, is to the authors of the programs and packages we describe. Every author whom we contacted to discuss problems provided us with practical help in the spirit of the TeX community, and often gave us permission to reuse examples from their documentation.

Christel Mittelbach substantially improved the music chapter and Leonor Barroca spent a great deal of time on the METAFONT and METAPOST chapter—

without their work we would never have finished this book.

We gratefully recognize all our many colleagues in the TeX world who developed the packages, not only those described here, but also the hundreds of others that help users typeset their documents faster and better. Without the continuous effort of all these enthusiasts, TeX would not be the magnificent and flexible tool it is today. We tried to do them justice by mentioning, when first describing a given package, the original author and/or other major contributors.

We are greatly indebted to Dwight Aplevitch, Barbara Beeton, Karl Berry, David Carlisle, Malcolm Clark, L. Peter Deutsch, Denis Girou, John Hobby, Alan Hoenig, Ross Moore, Thorsten Ohl, Kris Rose, Tom Rokicki, Chris Rowley, Virgil Stokes, Ulrik Vieth, and Timothy van Zandt for their careful reading of various chapters of the manuscript. Their numerous comments, suggestions, corrections, and hints have substantially improved the quality of the text.

On the publishing side, we wish to thank Peter Gordon, our editor at Addison Wesley Longman, Inc., who gave us much-needed support and encouragement over the three years duration of this project. When it came to production, Avanda Peters and Rosa Gonzalez were unfailingly patient with our idiosyncrasies and steered us safely to completion. Katrina Avery edited our dubious prose into real English, and Art Ogawa took great care in overseeing the typesetting and the color pages; we greatly appreciate their work.

Feedback

We would like to ask you, dear reader, for your collaboration. We kindly invite you to send your comments, suggestions, or remarks to any of the authors. We shall be glad to correct any mistakes or oversights in a future edition, and are open to suggestions for improvements or the inclusion of important developments we may have overlooked. We will maintain a list of errata in a file called `grphcomp.err` in the \LaTeX distribution, and this will contain current addresses for the authors.

Loading P₁CT_EX without problems

a useful hack for L^AT_EX users

Hans Hagen

April 1 1997

T_EX provides 256 *<dimensions>* and 256 *<skips>*. In ConT_EXt this is no problem, but in packages that have many authors, one can be quite sure that a lot of *<dimensions>* are allocated. Packages that use P₁CT_EX can therefore run out of *<dimensions>* quite fast. This module was written as a reaction to persistent problems with loading PPCHT_EX in L^AT_EX and P₁CT_EX deserves a solution. I therefore dedicate this module to Tobias Burnus and Dirk Kuypers, who use PPCHT_EX in a L^AT_EX environment and suggested a lot of extensions to the repertoire of PPCHT_EX commands.

This module presents a solution that is quite effective: all *<dimensions>* are drawn from the pool of *<dimensions>* and *<skips>*, depending on the availability. This is possible because *<dimensions>* are *<skips>* without a glue component. Therefore we can use *<skips>* as *<dimensions>*. However, some incompatibility can result from assignments that look like:

```
\somedimen=\someskip
```

In such cases the *<dimension>* equals the fixed part of the *<skip>* or in other words: this assignment strips off the glue. Because P₁CT_EX uses no glue components, I thought I could interchange both register types without problems, but alas, this didn't hold for all *<dimensions>*.

In Plain T_EX the allocation macros are defined with (as) `\outer`. This means that they cannot appear inside macros, not even in an indirect way. We therefore have to redefine both `\newdimen` and `\newskip` to non-`\outer` alternatives. In most macro packages this redefinition already took place. We save the original meanings, so we can restore them afterwards.

```
1 \let\normalnewdimen = \newdimen
  \let\normalnewskip = \newskip
2 \catcode'@=11 % I'd rather used \unprotect
  \def\temporarynewdimen {\alloc@1\dimen\dimendef\insc@unt}
  \def\temporarynewskip {\alloc@2\skip \skipdef \insc@unt}
  \catcode'@=12 % and \protect.
```

Here comes the trick. Depending on how many *<dimensions>* and *<skips>* are allocated, the `\newdimen` assigns a *<dimensions>* or *<skip>*. Plain T_EX allocates 15 *<dimensions>* and 17 *<skips>*. After loading P₁CT_EX, 71 *<dimensions>* and 71 *<skips>* are allocated. Indeed, P₁CT_EX needs 110 *<dimensions>*!

```
\def\newdimen%
  {\ifnum\count11>\count12
   \let\next=\temporarynewskip
  \else
   \let\next=\temporarynewdimen
  \fi
  \next}
```

When I was testing a new version of PPCHT_EX in Plain T_EX I had to find out that this exchange of registers sometimes leads to unwanted results. It took me some hours to find out that the source of errors originated in constructions like:

```
\ifdim\DimenOne<\DimenTwo whatever you want \else or not \fi
```

When `\DimenOne` is a *<skip>* and `\DimenTwo` is a *<dimension>*, T_EX scans for some optional glue component, like in:

```
\skip0=\dimen0 plus 10pt minus 5pt
```

The most robust solution to this problem is:

```
\ifdim\DimenOne<\DimenTwo\relax right \else wrong \fi
```

Some close reading of the P_{CT}E_X source however learned me that this problem could be solved best by just honoring the allocation of *<dimensions>* when the name of the macro explicitly stated the character sequence *dimen*. A next implementation therefore automatically declared all *<dimensions>* with this sequence in their names with `\dimen`. Again I was too optimistic, so now we do it this way (the comments are from P_{CT}E_X, which like T_AB_LE, is an example of a well documented package):

```

3 \catcode'!=11
\temporarynewdimen\!dimenA      %.AW.X.DVEUL..OYQRST
\temporarynewdimen\!dimenB      %...X.DVEU...O.QRS.
\temporarynewdimen\!dimenC      %..W.X.DVEU.....RS.
\temporarynewdimen\!dimenD      %..W.X.DVEU...Y.RS.
\temporarynewdimen\!dimenE      %..W.....G..YQ.S.
\temporarynewdimen\!dimenF      %.....G..YQ.S.
\temporarynewdimen\!dimenG      %.....G..YQ.S.
\temporarynewdimen\!dimenH      %.....G..Y..S.
\temporarynewdimen\!dimenI      %..BX.....Y....
\temporarynewdimen\!dxpos       %..W.....U..P...S.
\temporarynewdimen\!dypos       %..WB.....U..P.....
\temporarynewdimen\!xloc        %..WB.....U.....S.
\temporarynewdimen\!yloc        %.....L.P..Q.ST
\temporarynewdimen\!ypos        %.....L.P..Q.ST
\temporarynewdimen\!zpt         %.AWBX.DVEULGP.YQ.ST

```

Tobias tested this module in all kind of L^AT_EX dialects so we were able to find out that we also needed to declare:

```

4 \temporarynewdimen\linethickness
\catcode'!=12

```

After all, the new definition of `\newdimen` became:

```

5 \def\newdimen#1%
  {\ifx#1\undefined
   \ifnum\count11>\count12
    \temporarynewskip#1\relax
   \else
    \temporarynewdimen#1\relax
   \fi
  \fi}

```

Curious readers can still find the previous solution in the source. The next macro is used instead of `\input`. This macro also reports some statistics.

```

6 \def\dimeninput#1 %
  {\message{[before: d=\the\count11,s=\the\count12]}}%
  \input #1 \relax
  \message{[after: d=\the\count11,s=\the\count12]}}%

```

Not every package defines `\fiverm`, P_{CT}E_X's pixel, so let's take care of that omission now:

```

7 \ifx\undefined\fiverm
  \font\fiverm=cmr5
\fi

```

The actual loading of P_{CT}E_X depends on the package. For L^AT_EX users we take care of loading the auxiliary ones too.

```

8 \ifx\beginpicture\undefined
  \ifx\newenvironment\undefined
    \dimeninput pictex \relax
  \else
    \dimeninput prepicte \relax
    \dimeninput pictex \relax
    \dimeninput postpict \relax
  \fi
\fi

```

Finally we restore the old definitions of `\newdimen` and `\newskip`:

```
9 \let\newdimen = \normalnewdimen
  \let\newskip = \normalnewskip
  and just hope for the best.
10 \endinput
```

PDF \TeX , een eerste indruk

Hans Hagen

April 1 1996

De aanleiding

Enkele weken terug vroeg Erik Frambach me of ik bereid was de web2c \TeX omgeving te testen. Deze omgeving zal mogelijk de kern vormen van 4 \TeX voor Windows (of hoe dat ook mag heten). Het installeren van deze hi- \TeX omgeving was vrij eenvoudig, met name omdat er slechts één installatiefile is. We kunnen op dit moment binnen onze organisatie dus kiezen uit HugeEm \TeX (dat DOS ondersteunt), YandY \TeX (dat, dankzij het dynamisch geheugenbeheer, het gelijktijdig draaien van meerdere jobs mogelijk maakt) en Web2c \TeX (waarvan als enige ook de source beschikbaar is).

In eerste instantie installeerde ik alleen de meest relevante delen van deze distributie: het programma zelf, de pool file en wat DLL-files. Aangezien wij vrij veel PDF output produceren, drukte ik ook wat documentatie van de variant PDF \TeX af. Bestudering hiervan was reden genoeg om ook dit programma maar eens te installeren.

PDF \TeX , wat is dat?

Knuth heeft, sinds hij \TeX en METAFONT ontwikkelde, regelmatig geschreven en gezegd dat zijn versie weliswaar bevroren is, maar dat anderen er naar hartelust aan mogen sleutelen. De afgelopen jaren hebben slechts weinigen deze uitdaging aanvaard.

John Hobby heeft bijvoorbeeld METAFONT zodanig aangepast en uitgebreid, dat geavanceerde POSTSCRIPT output mogelijk is. Vanuit het oogpunt van de gebruiker is dit wellicht de meest belangwekkende aanpassing van een van Knuth's programma's. Ik zou zeggen, werp eens een blik op MetaPost.

Op het \TeX front hebben de ontwikkelingen zich vooral beperkt tot uitbreidingen die eerder format-schrijvers dan gebruikers zullen plezieren. Zo is er al enige tijd de multilinguale duizendpoot Omega en is er sinds kort e- \TeX , waarmee ik binnenkort aan de slag hoop te gaan.

Het door Han The Thanh, Petr Sojka & Jiří Zlatuška ontwikkelde PDF \TeX is voor \TeX , wat MetaPost is voor METAFONT. Waar MetaPost aansluiting realiseerde bij het (inmiddels tot standaard verheven) POSTSCRIPT, zorgt PDF \TeX voor aansluiting op het (wellicht op korte termijn tot standaard te verheffen) PDF formaat.

Het traject om van een in \TeX gecodeerde brontekst tot PDF te komen, loopt via DVI en POSTSCRIPT. Met de komst van PDF \TeX komen twee slagen te vervallen. Dit

betekent niet alleen een winst in termen van verwerkingstijd, het scheelt ook schijfruimte. Bovendien neemt de kans af dat problemen optreden tijdens het vertalen naar PDF. Dit laatste aspect moet niet worden verwaarloosd. Zo zitten er in de recente versie van Acrobat Distiller een paar storende fouten, die op deze manier goed te omzeilen zijn. Een nog storender fout in Acrobat Reader dwingt me echter definitieve versie te verwerken met Distiller (level 2.1 code).

De functionaliteit

Samengevat komt het er dus op neer dat PDF \TeX in plaats van DVI files PDF files genereert. Nu zal een kritische lezer zich inmiddels al hebben afgevraagd of alles wat we tot nu toe kon met \TeX ook met PDF \TeX kan. Immers, veel macropakketten gebruiken de \TeX primitieve `\special` om effecten te bewerkstelligen die \TeX niet in zich heeft:

- het integreren van illustraties (bitmap en vector)
- het weergeven van (tekst in) kleur
- roteren, ronde hoeken en andere grafische geintjes
- het ondersteunen van hypertextmogelijkheden

We kunnen potentiële gebruikers geruststellen: veel is mogelijk.

Men kan bijvoorbeeld figuren integreren met behulp van de primitieve `\pdfimage`. De reikwijdte van deze primitieve is echter beperkt tot files in het bitmap PNG formaat. Andere bitmap formaten kunnen echter vooraf worden omgezet in dit formaat.

Anders ligt het met vector graphics. Aangezien de eerste processlag meteen de laatste is, kunnen we deze niet zoals gebruikelijk afhandelen met specials. Specials hebben bij afwezigheid van postprocessing eigenlijk hun functie verloren. De primitieve `\special` is dan ook gereduceerd tot een die alleen tussenvoegen van PDF code toestaat. Aangezien met GhostView figuren in EPS formaat kunnen worden omgezet in PDF formaat, kunnen toch dergelijke figuren worden tussengevoegd. Hier blijkt het nut van een discussielijst, waarop Tanmoy Bhattacharya uit de doeken deed hoe een door GhostView geproduceerde file kon worden omgezet in een PDF file die door een utility wordt omgezet in een PDF \TeX file, die op haar beurt weer met `\input` kon worden tussengevoegd (een doordenkertje). Wat experimenteren leerde me dat de tweede tussenstap niet echt nodig is en dat \TeX het heel goed zelf afkan, en nog snel ook. Echter, niet alle door GhostView geproduceerde output is op dit moment geschikt. Voor hen die

dat nog niet weten: GhostView is (en wordt) een uitstekend alternatief voor Acrobat Reader!

De tot PDF special gereduceerde `\special` kan ook worden gebruikt om kleuren in te stellen. En aangezien T_EX uitstekend in staat is zelf het kleurmanagement te voeren, is ondersteuning van kleur dus ook geen probleem.

Wat lastiger ligt het met specifieke grafische mogelijkheden. Waar POSTSCRIPT een volwassen programmeertaal is, is PDF slechts een weergavetaal. Dit betekent dat we in PDF bijvoorbeeld niet kunnen rekenen. Het PDF producerende programma moet dus het rekenwerk doen, en dat is in dit geval dus T_EX. Het implementeren van bijvoorbeeld kaders met ronde hoeken is nauwelijks een probleem, evenals roteren. Geavanceerde macropakketten als PStricks zullen echter grotendeels moeten worden herschreven, tenzij men natuurlijk de plaatjes afzonderlijk produceert en omzet in PDF. Het aanpassen van MetaPost danwel het omzetten van door dit programma geproduceerde output zou nauwelijks problemen moeten opleveren, maar op dit moment expandeert 5K POSTSCRIPT nog tot 633K PDF, dat door PDF_TE_X weer wordt gecomprimeerd tot 120K.

PDF_TE_X gaat ver in de ondersteuning van hypertext mogelijkheden. Er zijn daartoe een aantal nieuwe primitieven beschikbaar —de PDF_TE_X primitieven beginnen allemaal met `\pdf`— die een curieuze mix vormen van T_EX en verbatim PDF. Het gebruik van de primitieven `\pdfdest` en `\pdfannotlink`, iets dat de meeste macropakketten verbergen voor hun gebruikers, vergt dus wel wat kennis van PDF.

Een zeer groot voordeel van PDF_TE_X is dat het programma nog steeds DVI-code kan produceren. Persoonlijk houd ik graag dit achterdeurtje open, niet in de laatste plaats omdat het uit DVI te produceren POSTSCRIPT vaak veel beter overdraagbaar is dan PDF, wat vooral een gevolg is van tekortkomingen in viewers. Bovendien is DVI output vaak prettig compact en zijn DVI viewers veel beter geschikt om te testen: ze starten sneller op en zijn in weergave superieur.

De keuze tussen PDF en DVI wordt bepaald door de primitieve `\pdfoutput`. Aanzienlijk compactere files worden verkregen met behulp van `\pdfcompresslevel` en `\pdfsubset`. Dit laatste benadrukt nog eens een essentieel verschil met DVI: de fonts zitten in de file!

Het aanpassen van macropakketten

De vraag zal wellicht rijzen hoe eenvoudig een beetje macropakket is aan te passen aan PDF_TE_X. Ik kan zeggen: dat valt reuze mee. Het heeft me precies een dag gekost om een interface te schrijven die de functionaliteit van ConT_EXt ondersteunt. We praten dan over: kleur, kaders met ronde hoeken, het tussenvoegen van plaatjes, hyperlinks in alle toonaarden en wat opsmuk voor interactieve teksten, zoals full screen opstarten (iets dat onder Distiller 3 geheid tot problemen leidt). De documentatie was helaas zeer gebrekkig, maar de WEB-source bood gelukkig uitkomst.

Binnen ConT_EXt zijn alle specials ondergebracht in aparte modules. Ik hoefde dus alleen een nieuwe module toe te voegen. Om een indruk te krijgen van de complexiteit van zo'n module en de aard van de aanpassingen, geven we hieronder een deel van de betreffende module weer:

```
\startspecials[tpd]

\definespecial\dostartgraymode#1%
{\special{PDF: #1 g}}

\definespecial\dostopgraymode%
{\special{PDF: 0 g}}

\definespecial\dostartcolormode#1#2#3%
{\special{PDF: #1 #2 #3 rg}}

\definespecial\dostopcolormode
{\special{PDF: 0 0 0 rg}}

\definespecial\dostartrotation#1%
{\processaction
 [#1
 [ 90=>\special{PDF:q 0 1 -1 0 0 0 cm},
 180=>\special{PDF:q -1 0 0 -1 0 0 cm},
 270=>\special{PDF:q 0 -1 1 0 0 0 cm},
 360=>\special{PDF:q 1 0 0 1 0 0 cm}]]}

\definespecial\dostoprotation%
{\special{PDF:Q}}

\definespecial\dosetupinteraction%
{\pdfoutput=1\relax
 \pdfcompresslevel=9\relax
 \pdfsubset=1\relax}

\definespecial\dostartthisislocation#1%
{\pdfdest name {#1} fitp}}

\definespecial\dostartgotolocation#1#2#3#4#5%
{\doifelse{#3}{\jobname}
 {\!!doneafalse}
 {\doifelse{#3}{}
 {\!!doneafalse}
 {\!!doneatruer}}}%
 \pdfannotlink
 width #1sp
 height #2sp
 depth 0pt
 attr{/Border [ 16 16 0 ]}
 goto \if!!donea file {#3.pdf} \fi name {#4}%
 \pdfendlink}

\definespecial\dosetupscreen#1#2#3#4#5%
{\pdfpagewidth =#3sp
 \pdfpageheight=#4sp
 \ifcase#5\else
 \pdfcatalog {/PageMode /FullScreen}%
 \fi}
```

```
\definespecial\dosetupidentity#1#2#3#4#5%
{\pdfinfo
  /Title (#1)
  /Subject (#2)
  /Author (#3)
  /Creator (#4)
  /ModificationDate (#5)}}

```

```
\stopspecials
```

De niet getoonde macros betreffen het tussenvoegen van externe files (figuren), het tekenen van kaders met ronde hoeken en wat additionele interactiviteit.

Binnen Con \TeX t kunnen specials exclusief of parallel worden uitgevoerd. De PDF \TeX specials moeten per se exclusief zijn, dat wil zeggen dat er niet gelijktijdig specials voor andere drivers mogen worden weggeschreven. De switch van DVI naar PDF output wordt dan ook gemaakt met:

```
\gebruikspecials[reset, tpd]
```

Ik kan van deze plaats niet zeggen hoe eenvoudig het is om een willekeurig macropakket aan te passen. Veel hangt af van de wijze waarop specials worden ondersteund. Her en der door de source verspreide `\specials` maken het er in ieder geval niet eenvoudiger op.

Enkele kanttekeningen

Is hiermee DVI overbodig geworden? Zoals ik al aangaf, blijven we natuurlijk DVI gebruiken. Tijdens het schrijven van deze tekst maak ik gebruik van DVI previewers (van E.Mattes en YandY) die ik geen van beiden zou willen missen. Hetzelfde geldt voor de hoogwaardige POSTSCRIPT output.

Om TIFF en EPS figuren te kunnen opnemen moet men eerst de figuren converteren met GhostView. Op dit moment is die conversie niet optimaal, zowel in termen van kwaliteit als in termen van omvang van de files. Hoe lang zal het echter nog duren voordat alle tekenpakketten files ook in PDF formaat kunnen exporteren? Hoe dan ook: het hoort niet tot de taken van \TeX dit te doen. In die zin is de primitieve `\pdfimage` dan ook eigenlijk overbodig. De kunst is nu nog de bounding box informatie uit de tussen te voegen PDF file te destilleren.

Het afhandelen van verbatim POSTSCRIPT valt natuurlijk buiten de scope van PDF \TeX . Specials zijn niet voor niets specials. In die zin hadden de ontwerpers beter een nieuwe primitieve `\pdfspecial` kunnen implementeren en `\special` kunnen gebruiken in de traditionele zin, namelijk het tussenvoegen van verbatim code in de vorm van commentaar.

Ik ben hierboven voorbij gegaan aan enkele andere mogelijkheden. Zo ondersteunt PDF \TeX het aanmaken van outlines. Dit lijken me voor \TeX gebruikers nauwelijks nodig, omdat de meeste wat omvangrijker documenten zelf zorg dragen voor hun inhoudsopgave. Waarom zouden we hoogwaardige typografie inleveren voor een door een viewer gegenereerde inhoudsopgave, die bovendien nog een deel van het scherm in beslag neemt.

PDF \TeX ondersteunt ook het hergebruik van informatie (meestal graphics), of in PDF termen: form objects. Daarnaast kunnen article treads worden aangemaakt. Dit laatste is vooral handig wanneer we lezers door een tekst willen loodsen. Tot slot noemen we nog de mogelijkheid om annotaties op te nemen. Natuurlijk heb ik nog wel een verlanglijstje.

Even voorstellen: de EPDA

Ton Otten

April 1 1996

Inleiding

In juni 1996 hebben enkele organisaties het initiatief genomen tot de oprichting van de European Portable Document Association (EPDA). In termen van de *N^TG* had die naam ook kunnen luiden: PDF-gebruikersgroep.

Enige jaren geleden heeft Adobe Systems Inc. het Portable Document Format (PDF) en de bijbehorende Acrobat-programmatuur geïntroduceerd. Dit formaat en de programmatuur maakt platformafhankelijke verspreiding van elektronische documenten mogelijk. In eerste instantie was deze programmatuur bedoeld voor pre-press doeleinden, maar op dit moment worden steeds meer PDF-toepassingen ontwikkeld, waarbij PDF als distributiemedium wordt ingezet.

Door de toenemende belangstelling voor elektronische informatievoorziening, gecombineerd met zeer specifieke wensen ten aanzien van vormgeving en kwaliteit, is er een grote behoefte aan kennis rond het PDF. Die kennis is nog zeer verdeeld in het bedrijfsleven aanwezig.

Om de inspanningen van de verschillende bedrijven enigszins te bundelen is de EPDA opgericht voor die mensen die als PDF-producent of -ontwikkelaar op professionele wijze bezig zijn met PDF.

Ook T_EX-gebruikers zijn geïnteresseerd in PDF als een *output format* naast bijvoorbeeld DVI en PostScript. Die interesse komt bijvoorbeeld goed tot uiting in de ontwikkeling van PDF_TE_X. Elders in dit nummer schrijft mijn collega over zijn eerste ervaringen met PDF_TE_X.

De leden

De EPDA is bedoeld voor die bedrijven, organisaties, instellingen en/of individuen die zich in de breedste zin van het woord bezighouden met PDF en/of de produktie en distributie van elektronische documenten.

De doelen

De doelstellingen van de EPDA kunnen als volgt worden geformuleerd:

- De EPDA conformeert zich aan PDF als standaard voor de distributie van elektronische documenten en bevordert het gebruik van PDF.
- De EPDA behartigt de collectieve belangen van de leden door het op (inter)nationaal niveau onderhouden van contacten met organisaties of instellingen (Adobe, gebruikersgroepen, overheid enz.) die een rol spelen in de ontwikkeling van PDF.
- De EPDA vormt voor haar leden een communicatieplatform waarop deze onderling en met derden informatie met betrekking tot PDF kunnen uitwisselen.

- De EPDA stimuleert de standaardisatie van PDF tot een algemeen aanvaarde bedrijfsstandaard cq. ISO-norm.
- De EPDA is een aanspreekpunt voor derden die geïnformeerd willen worden over PDF.
- De EPDA stimuleert de totstandkoming van een sluitende regelgeving rond de auteursrechterlijke kant van de elektronische distributie van documenten.
- De EPDA stelt zich ten doel de kwaliteit van PDF-documenten te waarborgen.

Bij het lezen van deze doelstellingen moet er rekening mee worden gehouden dat er naast PDF ook andere ontwikkelingen gaande zijn op het gebied van de standaardisering van de platformafhankelijke verspreiding van elektronische documenten. Te denken valt aan nieuwe formaten die tot standaard worden verheven. Deze ontwikkelingen worden door de EPDA nauwlettend gevolgd, hetgeen eventueel tot aanpassing van deze doelstellingen kan leiden.

De organisatie

Om de doelen te kunnen realiseren is de EPDA georganiseerd in twee afdelingen:

- Research & Application
Deze afdeling heeft tot taak oplossingen te vinden voor de technische problemen rond PDF, ontwikkelingen te volgen en contacten te onderhouden met de technische afdelingen van Adobe Systems Inc.
- Communication
Deze afdeling verzorgt de interne en externe communicatie. Bovendien draagt zij zorg voor de verdere ontwikkeling van de EPDA (promotie).

Onder verantwoordelijkheid van het bestuur worden diverse projecten uitgevoerd. Een van die projecten heeft tot doel PDF tot een ISO-norm te maken, opdat PDF als een bedrijfsstandaard kan worden geaccepteerd. Hiermee wordt voorkomen dat PDF onder invloed van marktontwikkelingen teveel aan wijzigingen onderhevig zal zijn. Het project heeft een looptijd van twee jaar en wordt uitgevoerd door de EPDA, de KUB, het NNI en TNO.

EPDA
Postbus 125
8000 AC Zwolle
telefoon: +31 (0)38 422 29 60
internet: in voorbereiding
www-pagina: in voorbereiding

Puzzling Graphics in METAPOST

Hans Hagen

April 1 1996

In the beginning

If all stories are true, \TeX is widely used among mathematicians and is not that popular among what we call α -oriented people. Maybe that's due to the command like interface of this system, although well designed macro packages could be of great help. Or does mathematical oriented use inhibit transfer to other domains.

When browsing through periodicals of user groups, one seldom finds articles on the use of METAFONT other than for making fonts. How can it be that people sometimes use \TeX for non \TeX nical purposes (like drawing things and calculating other things) while they oversee METAFONT as a tool that suits many drawing purposes other than fonts. And, every time I see some output, the quality strikes me.

Being no real mathematician I never came to reading the whole METAFONT book. I have generated some fonts, but that was all. Out of frustration about the quality of some drawing packages, I could not resist installing MetaPost, the POSTSCRIPT producing ancestor of METAFONT.

However, MetaPost nearly drove me crazy, when I found out that including \TeX output is possible, but only in close cooperation with tuned DVI to POSTSCRIPT drivers, unfortunately not the one I used. This forced me to write some \TeX hacks first, but finally I was able to produce MetaPost output which included text typeset by \TeX . For those who want to know the details: this dependancy concerns the inclusion of fonts. MetaPost expects DVI to POSTSCRIPT converters to sort this out, but one can trick \TeX to force this behavior.¹ Another dependency surfaced a while later: the macropackage one uses must leave \TeX 's vertical and horizontal offsets untouched, i.e. at it's initial values of 1 inch.

Visual proofing

Actually using MetaPost was stimulated by an (indeed) mathematical problem someone presented me. It took me just a few minutes to see that a sort of graphical proof was far more easy than some prove with a lot of $\sin(\textit{this})$ and $\cos(\textit{that})$ and long forgotten formulas.

I was expected to prove that the angle between the lines connecting the mid points of four squares drawn upon the four sides of a box is ninety degrees (see figure)

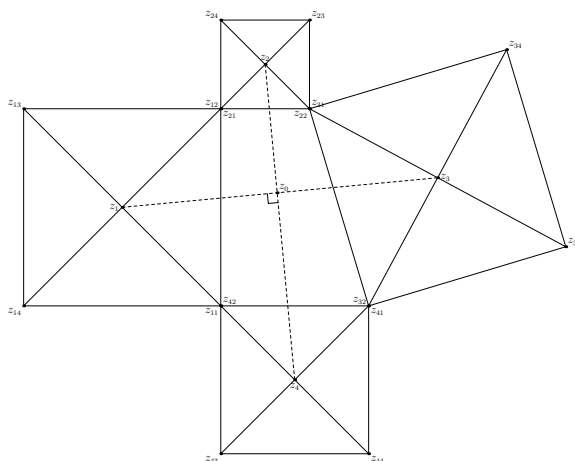


Figure 1: The problem.

¹The module `m-metapost` handles those font matters.

When I was drawing the extremes of this graphic, I could not resist starting up the computer and setting MetaPost to work. And indeed, I had to find out that programming in this language is not that hard and sometimes is even more fun than programming in \TeX .

Looking at the program and the graphics it produces, it looked, to me, that at least in educational settings it should be far more easy to convert people to \TeX /MetaPost showing them MetaPost than showing them some bulky and complicated \TeX macropackage.

I will explain a few details of the next MetaPost program, which probably holds no beauty for real advanced users. First we enable generation of some prologue material. This is needed when we include typeset objects. I still wonder why this is option not turned on by default; MetaPost files are always very small compared to those generated by drawing packages and the extra overhead is minimal.

```
1 prologues := 1;
```

For the sake of this article I've added some labels to the points we use. Generating these label however is optional and controlled by a boolean:

```
2 boolean show_labels; show_labels := false;
```

We hide the problem in a macro, which accepts four pair of coordinates that determine the central four sided shape.

```
3 def draw_problem (expr p, q, r, s ) =
```

Because we want to call this macro more than once, we have to save the locally used values. Instead of declaring local variables, one can hide their use for the outside world. In most cases variables behave global. If we don't save them, the next call will lead to errors due to conflicting equations.

```
4   save x, y, a, b, c, d, e, f, g, h;
```

We draw four squares and make use of MetaPost's powerful capability to solve equations. Watch the use of = which means that we just state dependencies. Assignments are realized by :=. The less assignments we use, the less errors we can make.

```
5   z11 = p   ; z12 = q   ; z22 = r   ; z32 = s   ;
   z31 = z22 ; z21 = z12 ; z41 = z32 ; z42 = z11 ;
```

```
6   a = x12 - x11 ; b = y12 - y11 ;
```

```
7   z11 = (x11, y11) ; z12 = (x12, y12) ;
   z13 = (x12-b, y12+a) ; z14 = (x11-b, y11+a) ;
```

```
8   c = x22 - x21 ; d = y22 - y21 ;
```

```
9   z21 = (x21, y21) ; z22 = (x22, y22) ;
   z23 = (x22-d, y22+c) ; z24 = (x21-d, y21+c) ;
```

```
10  e = x32 - x31 ; f = y32 - y31 ;
```

```
11  z31 = (x31, y31) ; z32 = (x32, y32) ;
   z33 = (x32-f, y32+e) ; z34 = (x31-f, y31+e) ;
```

```
12  g = x42 - x41 ; h = y42 - y41 ;
```

```
13  z41 = (x41, y41) ; z42 = (x42, y42) ;
   z43 = (x42-h, y42+g) ; z44 = (x41-h, y41+g) ;
```

So far for the calculations to be done. It is possible to trace the way MetaPost solves its equations. Very handy indeed! Now we can draw the lines, but first we choose ourselves a decent linethickness.

```
14  pickup pencircle scaled .5pt;
```

```
15  draw z11--z12--z13--z14--z11; draw z11--z13; draw z12--z14;
   draw z21--z22--z23--z24--z21; draw z21--z23; draw z22--z24;
   draw z31--z32--z33--z34--z31; draw z31--z33; draw z32--z34;
   draw z41--z42--z43--z44--z41; draw z41--z43; draw z42--z44;
```

The mid points can be calculated by MetaPost too. The next lines state that those points lay halfway the extremes.

```
16  z1 = 0.5[z11,z13] ; z2 = 0.5[z21,z23] ;
   z3 = 0.5[z31,z33] ; z4 = 0.5[z41,z43] ;
```

We choose a alternative linestyle to show them in full glory.

```
17 draw z1--z3 dashed evenly;
    draw z2--z4 dashed evenly;
```

Just to be complete, we add a symbol that marks the right angle. First we determine the common point of the two lines, that lays at whatever point MetaPost finds suitable.

```
18 z0 = whatever[z1,z3] = whatever[z2,z4];
    mark_rt_angle (z1, z0, z2) ;
```

The macro `mark_rt_angle` actually draws the angle symbol.

Next come the labels. This bunch of repetitive macros can surely be written more compact when we use an auxiliary macro. But manipulating strings and tokens is yet behind my experience and except when writing macro-packages there is no real need for this. To keep things readable we did not optimize the positioning.

```
19 if show_labels:
```

We use a bigger pen for drawing the dots.

```
20 pickup pencircle scaled 5pt;
```

The next two lines show that \TeX can do some typesetting work. We could have done without \TeX , but why not go for the best.

```
21 verbatimex \def\Z #1{\scriptscriptstyle #1}$} etex;
    verbatimex \def\ZZ#1#2{\scriptscriptstyle #1#2}$} etex;
```

The macro `dotlabel` draws a dot and puts a label. The `btex .. etex` construction moves control to \TeX . This data is processed afterwards and the results are used in the next run. Including fonts is a bit tricky, because it's DVI-driver dependant. To prevent problems we secretly include the used glyphs in the \TeX source and that way fool our DVI to POSTSCRIPT driver to include them. The (original) source of this text therefore includes:

```
\usemodule[metapost]          % a standard ConTeXt command
\UseMetaPostFile{filename}    % a not yet embedded module primitive
\UseMetaPostProofFont{cmr10} % another module primitive
```

This module is independent of Con \TeX t and is available for those who need it. So if the labels don't look like z_{12} , the wrong production tools were used. But let's switch back to the labels now:

```
22 dotlabel.llft(btex \ZZ 11 etex, z11); dotlabel.ulft(btex \ZZ 12 etex, z12);
    dotlabel.ulft(btex \ZZ 13 etex, z13); dotlabel.llft(btex \ZZ 14 etex, z14);
23 dotlabel.lrt (btex \ZZ 21 etex, z21); dotlabel.llft(btex \ZZ 22 etex, z22);
    dotlabel.urt (btex \ZZ 23 etex, z23); dotlabel.ulft(btex \ZZ 24 etex, z24);
24 dotlabel.urt (btex \ZZ 31 etex, z31); dotlabel.ulft(btex \ZZ 32 etex, z32);
    dotlabel.urt (btex \ZZ 33 etex, z33); dotlabel.urt (btex \ZZ 34 etex, z34);
25 dotlabel.lrt (btex \ZZ 41 etex, z41); dotlabel.urt (btex \ZZ 42 etex, z42);
    dotlabel.llft(btex \ZZ 43 etex, z43); dotlabel.lrt (btex \ZZ 44 etex, z44);
26 dotlabel.urt (btex \Z 0 etex, z0);
    dotlabel.lft (btex \Z 1 etex, z1); dotlabel.top (btex \Z 2 etex, z2);
    dotlabel.rt (btex \Z 3 etex, z3); dotlabel.bot (btex \Z 4 etex, z4);
```

And there we are.

```
27 fi;
28 enddef;
```

I copied the meaning of `mark_rt_angle` from the manual and I hope that someday I write such macros myself.

```
29 angle_radius = 10pt;
30 def mark_rt_angle (expr a, b, c) =
    draw ((1,0)--(1,1)--(0,1)) zscaled (angle_radius*unitvector(a-b)) shifted b
enddef;
```

We are going to draw a lot of pictures, so we define ourselves an extra macro. This time we hard-code some values. The fractions i and j are responsible for the visual iteration process.

```

31 def do_draw_problem ( expr n, i, j ) =
    beginfig ( n );
    draw_problem
      ( (400,400), (300,600),
        i[(300,600),(550,800)], j[(400,400),(550,500)] );
    endfig;
enddef;

```

Of course we could have used some loop here, but defining an auxiliary macro probably takes more time than simply calling the drawing macro directly. The results are shown on a separate page.

It does not need that much imagination to see the four sided problem converge to a three sided one, which itself converges to a two sided one. In the two sided alternative it's not that hard to prove that the angle is indeed 90 degrees.

```

32 do_draw_problem ( 40 , 1.0 , 1.0 ); do_draw_problem ( 41 , 0.8 , 1.0 );
do_draw_problem ( 42 , 0.6 , 1.0 ); do_draw_problem ( 43 , 0.4 , 1.0 );
do_draw_problem ( 44 , 0.2 , 1.0 ); do_draw_problem ( 45 , 0.0 , 1.0 );
33 do_draw_problem ( 30 , 0.0 , 1.0 ); do_draw_problem ( 31 , 0.0 , 0.8 );
do_draw_problem ( 32 , 0.0 , 0.6 ); do_draw_problem ( 33 , 0.0 , 0.4 );
do_draw_problem ( 34 , 0.0 , 0.2 ); do_draw_problem ( 35 , 0.0 , 0.0 );

```

We already showed a picture with coordinates. This picture was generated using some alternative coordinates.

```

34 def do_draw_problem ( expr n, i, j ) =
    beginfig ( n );
    draw_problem
      ( (400,400), (400,600),
        i[(400,600),(550,600)], j[(400,400),(550,400)] );
    endfig;
enddef;
35 show_labels := true;
36 do_draw_problem ( 50 , 0.6 , 1.0 );

```

Jigsawing

My second MetaPost project concerned our company logo, which I'll show in a few pages, when we combine it with the next one, which concerns jigsaw puzzles.

I still remember the times my parents and I spend whole evenings and sometimes part of the night solving jigsaw puzzles with thousands of pieces. The next piece of MetaPost code is therefore dedicated to my parents.

This kind of graphics are not that hard to generate with MetaPost, although MetaPost has to work real hard to draw real big ones.

Producing jigsaw originals is just a matter of laying a jigsaw pattern over some kind of picture. The macros we show here however support the generation of separate pieces, an option that we needed for producing a company postcard and a hypertext company profile, in which we use hot spot puzzle pieces.

First we generate the whole jigsaw puzzle. We feed the main macro with the width, the height and some unit.

```

37 def PreDrawPuzzle ( expr ww, hh, uu ) =

```

We use the MetaPost random generator. Because we want to be able to regenerate previous puzzles, the seed needs to be set explicitly.

```

38 randomseed := random_seed;
39 def rand (expr x) =
    ((x)+(uniformdeviate 10))/100
enddef;

```

We use the temporary variables w and h to store the width and height and also define our internal unit to be $uu/100$.

```

40 w := ww;
    h := hh;
    u := uu/100;

```

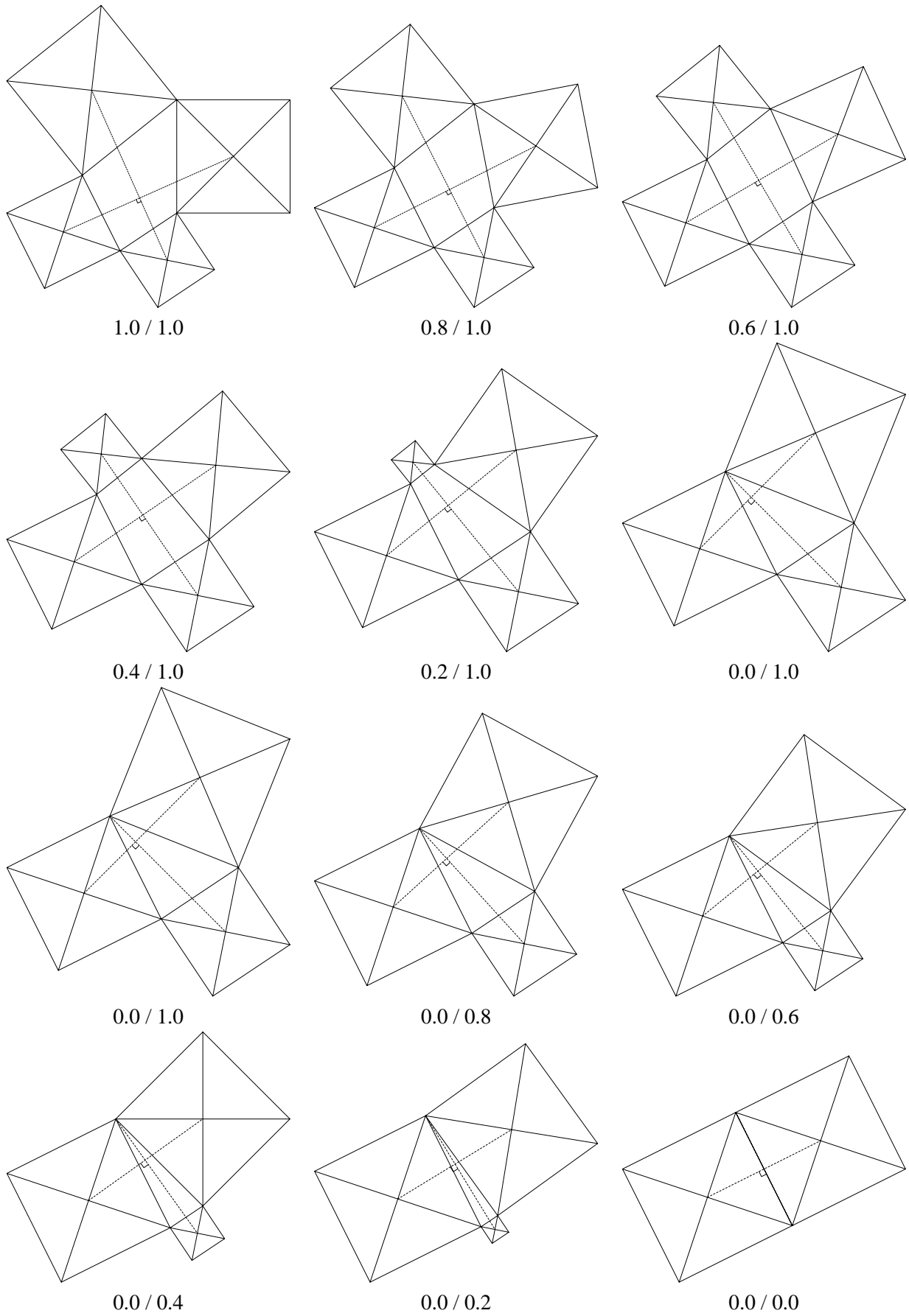


Figure 2: Some visual proof.

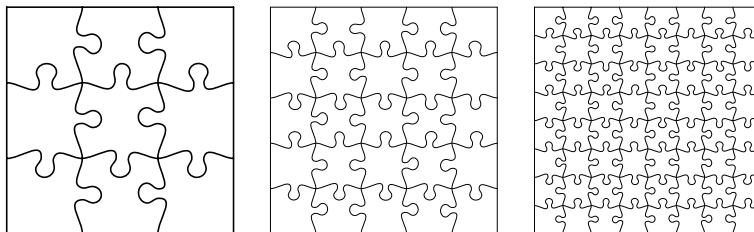


Figure 3: Some simple jigsaws.

The pieces of the puzzle show an alternating pattern. We just stick to this simple pattern, although some extra randomization can be used to generate some more randomized patterns. We use two boolean variables to keep track of the alternation:

```
41  boolean swapone; swapone := false;
    boolean swaptwo; swaptwo := false;
```

We will store the curves of the puzzle in two path variables, *pw* and *ph*, which hold the horizontal and vertical curves. We also need an auxiliary one. All three variables are arrays.

```
42  path p[], pw[], ph[];
```

The next local macro generates the path. At least some minimal knowledge of MetaPost is needed to understand them.

```
43  def buildpath (expr len, num) =
```

We fill *num* columns and/or rows with curves.

```
44  for j=1 upto num:
```

We need local variables and therefore call:

```
45  clearxy;
```

Next we take care of the alternation:

```
46  swaptwo := not swaptwo;
    if odd num:
        swapone := not swaptwo;
    else:
        swapone := swaptwo;
    fi;
```

The real work is done in the next few lines. We determine all relevant points on the curve. While doing so, the randomizer slightly offsets the curves from the midpoints. The values were determined experimentally.

```
47  for i=0 upto len+1:
    x[i]c = ((i*100)*u);
    y[i]c = ((j*100)*u);
    x[i]m = (x[i]c+50u);
    if swapone:
        y[i]m = (y[i]c-25u)
    else:
        y[i]m = (y[i]c+25u)
    fi;
    x[i]l = (rand(85)[x[i]c,x[i]m]);
    x[i]r = (rand(15)[x[i]m,x[i+1]c]);
    y[i]l = y[i]r=y[i]c;
    x[i]ll = (rand(75)[x[i]c,x[i]m]);
    x[i]rr = (rand(25)[x[i]m,x[i+1]c]);
    y[i]ll = y[i]rr=.50[y[i]c,y[i]m];
endfor;
```

We only used equations and let MetaPost do the calculations by solving those. Next we draw the path, and in doing this we have to take care of the begin and end points. These actually lay outside the puzzle boundaries, else the curves would end ugly.

```

48     p[j] :=
        for i=0 upto len:
            z[i]c..z[i]l..z[i]ll..z[i]m..z[i]rr..z[i]r..
        endfor
        z[len+1]c ;

```

Now that everything's done, we end the loop and the local macro definition.

```

49     endfor;
        enddef;

```

The boundaries of the puzzles are just straight lines.

```

50     pw[1] := (100u,100u)--(100u,(h+1)*100u);
        pw[w+1] := ((w+1)*100u,100u)--((w+1)*100u,(h+1)*100u);
51     ph[1] := (100u,100u)--((w+1)*100u,100u);
        ph[h+1] := (100u,(h+1)*100u)--((w+1)*100u,(h+1)*100u);

```

The next trickery uses the full strength of MetaPost: we clip of the parts of the curves that extend beyond the boundaries.

```

52     buildpath (w+1,h);
53     for k=2 upto h:
        ph[k] := p[k] cutbefore pw[1];
        ph[k] := ph[k] cutafter pw[w+1];
    endfor;
54     buildpath (h+1,w);
55     for k=2 upto w:
        pw[k] := p[w-k+2] rotated 90 shifted ((w+2)*100u,0);
        pw[k] := pw[k] cutbefore ph[1];
        pw[k] := pw[k] cutafter ph[h+1];
    endfor;

```

Drawing the whole puzzle can be done by flushing *pw* and *ph*, but as said before, we want to do some more. Therefore we save the puzzle piecewise in:

```

56     path phw[] [];

```

Here another powerful feature shows up. MetaPost itself determines where the pieces begin and end.

```

57     for kh=1 upto h:
        for kw=1 upto w:
            phw[kh][kw] := buildcycle (ph[kh], pw[kw], ph[kh+1], pw[kw+1]);
        endfor;
    endfor;

```

And now we're ready to use them.

```

58     enddef;

```

After the puzzle is predrawn, the horizontal curves are stored in *pw* and the vertical ones in *ph*. We also have the individual pieces available in *phw*. The most efficient way of drawing a whole puzzle is using *pw* and *ph*, like in:

```

59     def DrawWholePuzzle =
        for k=1 upto h+1:
            draw ph[k] withpen pencircle scaled 2.5u shifted (-100u,-100u);
        endfor;
        for k=1 upto w+1:
            draw pw[k] withpen pencircle scaled 2.5u shifted (-100u,-100u);
        endfor;
    enddef;

```

An individual piece can be output by the next macro, which takes two coordinates.

```

60     def DrawPuzzlePiece (expr ww, hh) =
        draw phw[ww][hh] withpen pencircle scaled 2.5u shifted (-100u,-100u);

```

enddef;

The boundaries are drawn by another macro:

```
61 def DrawPuzzleBorder =
    draw pw[1]--ph[1]--pw[w+1]--ph[h+1]--cycle
    withpen pencircle scaled 2.5u shifted (-100u,-100u);
enddef;
```

Now let's draw some puzzles first. We want to get the same ones every time, so we first set *random_seed* to some value. We'll give them the same dimensions, using *puzzle_units*.

```
62 puzzle_size = 3cm;
    random_seed := 10;

63 beginfig (101);
    PreDrawPuzzle (3, 3, puzzle_size/3);
    DrawWholePuzzle;
endfig;

64 beginfig (102);
    PreDrawPuzzle (5, 5, puzzle_size/5);
    DrawWholePuzzle;
endfig;

65 beginfig (103);
    PreDrawPuzzle (8, 8, puzzle_size/8);
    DrawWholePuzzle;
endfig;
```

As promised I'll show a more advanced application now, but first we need to load an illustration, like the following logo.

```
66 input mp-logo; separation := 0;
```

This one shows up as:

```
67 beginfig (201);
    PragmaLogo (1cm);
endfig;
```

And can be combined with the puzzle. Here we've realized the overlay in MetaPost, but one could as well let \TeX do the job.

```
68 beginfig (202);
    PreDrawPuzzle (3, 3, 4/3cm);
    DrawWholePuzzle;
endfig;

69 beginfig (203);
    PragmaLogo (1cm);
    PreDrawPuzzle (3, 3, 4/3cm);
    DrawWholePuzzle;
endfig;
```

Erasing a piece is quite straightforward. The next illustration can be used in tutorials on solving jigsaws.

```
70 def ErasePuzzlePiece (expr ww, hh) =
    erase fill phw[ww][hh] shifted (-100u,-100u);
enddef;

71 beginfig (204);
    PragmaLogo (1cm);
    PreDrawPuzzle (3, 3, 4/3cm);
    ErasePuzzlePiece (1, 1);
    DrawWholePuzzle;
endfig;
```

After we've drawn a picture, it's available in the MetaPost variable *currentpicture*, which means that we can use MetaPost's clip macro for clipping of a piece. Again, we defined ourselves a few auxiliary macro

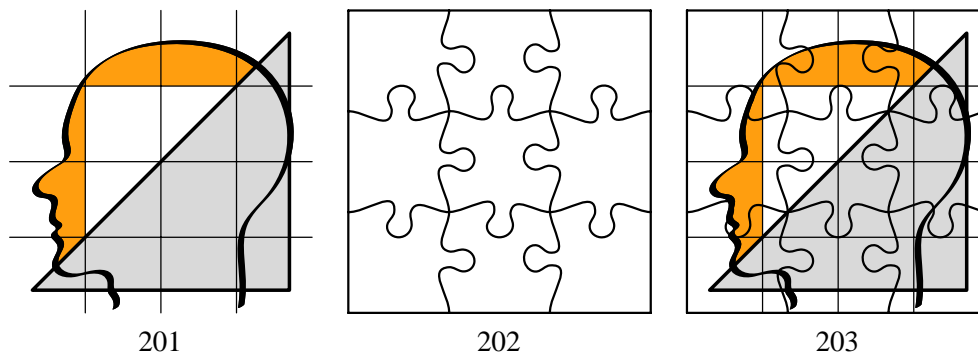


Figure 4: A real jigsaw.

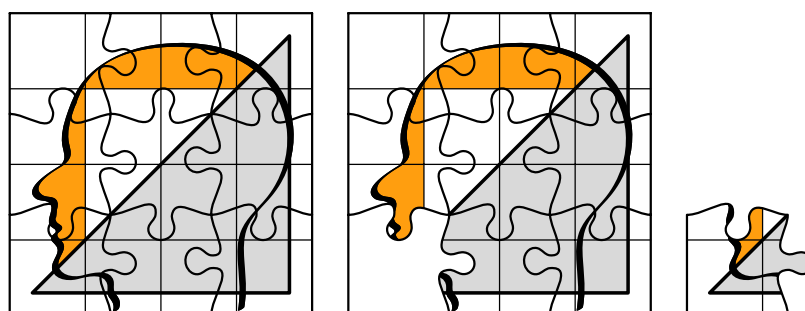


Figure 5: Some tutorial jigsaw.

```

72 def ClipToPuzzlePiece (expr ww, hh) =
    clip currentpicture to (phw[ww][hh] shifted (-100u,-100u));
enddef;
73 beginfig (205);
    PragmaLogo (1cm);
    PreDrawPuzzle (3, 3, 4/3cm);
    ClipToPuzzlePiece (1, 1);
    DrawPuzzlePiece (1, 1);
endfig;

```

Clipping is not as efficient as one would expect it to be. We can think of looking through a window to the puzzle: the whole picture is output, but only part of it is visible. In our third example we'll build a large picture out of clipped parts. The resulting files are quite large and take some time to print. A more efficient solution would be to let the printer calculate the non clipped picture once and reuse it, but as far as I can see, this is not supported in MetaPost. (There is no real need for such functionality anyway, because one seldom uses the same sub picture more than once.)

We used TeX to produce a company postcard, that showed 3 x 3 separate pieces. This involves some picture handling and therefore close reading of the MetaPost manual. The postcard was typeset using TeX. Because the postcard was in color, we also had to take care of color separation. The company logo is tuned to this which means that the colored areas are printed slightly under the black lines. Such extensions are not that hard to program once one knows the tricks.

Enlarging illustrations as the one described here are a good illustration of the quality of MetaPost. Using magnifying tools in viewers can be very cruel to designers, but not to MetaPost. Let's spend a few more words on accuracy.

When one looks at the wonderful compact output of MetaPost one will notice that coordinates are specified in high precision. The exact dimensions of the visible part of the picture are specified in the BoundingBox. Although probably no POSTSCRIPT interpreter prevents this, the bounding box is often specified in whole numbers. If one considers the very high accuracy of TeX on the one hand and the visible difference between a few POSTSCRIPT points, one can imagine that, especially when scaling is applied, visual imperfectness can surface. We found it quite frustrating that we had to move up an illustration by .5pt, just to get it's embedded background in line with one we let TeX produce in a menu button. After we studied the source of the illustration, we found out that it specified a HiResBoundingBox, so now we let TeX use this

one when it's there. Fortunately it's possible to let MetaPost output such a high resolution bounding box. Its accuracy deserves it!

Moving around

I don't even know how the next kind of gimmick is called, but let's name it a shift puzzle. We just show this example, which has not that much use (except as an alternative puzzling kind of pagENUMBERING symbol) because it shows how pictures can be manipulated in MetaPost.

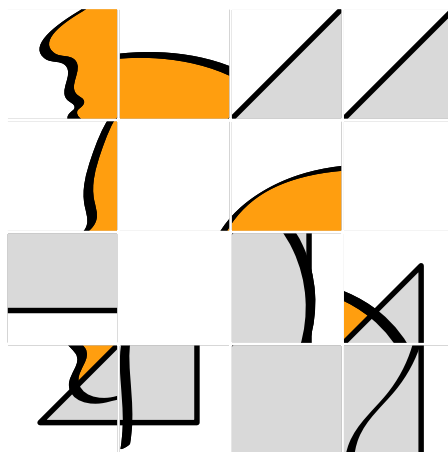


Figure 6: A shifting device.

Because we don't want to be dependant of dimensions, we first declare ourselves a decent unit u , and some height h and width w constant that we'll use later on. We also tell MetaPost how many figures we want to generate.

```
74 u := .75cm;
   w := 4;
   h := 4;
   n := 100;
```

```
75 input mp-logo; separation := 0; PragmaLogo (u);
```

After calling the macro, the picture is available in *currentpicture*. We are going to break this picture into smaller pieces, which we store in a 2-dimensional array of pictures:

```
76 picture r[] [];
```

We need some more variables, like a scratchpicture s :

```
77 picture s;
```

The next operation is needed because we are going to move pictures around. This kind of manipulations sometimes make designing graphics a nuisance.

```
78 currentpicture := currentpicture shifted (u,u);
```

Next we split off the subpictures that we are going to move around.

```
79 for step_w:= 1 upto w+1:
   for step_h:= 1 upto h+1:
     r[step_w][step_h] := currentpicture shifted (-step_w*u,-step_h*u);
     clip r[step_w][step_h] to (unitsquare scaled u);
   endfor;
endfor;
```

The auxiliary macro below takes care of generating a picture.

```
80 current_w := 1; prev_w := current_w;
   current_h := h; prev_h := current_h;
```

```
81 boolean cut_pieces; cut_pieces := false;
```

```
82 def generate_picture ( expr fig_number ) =
    beginfig (fig_number);
```

First we clear anything left.

```
83 clearit;
```

This double loop builds a large picture out of the clipped ones we saved earlier.

```
84 for step_w:= 1 upto w:
    for step_h:= 1 upto h:
        addto currentpicture also ( r[step_w][step_h] shifted (step_w*u,step_h*u));
    endfor;
endfor;
unfill unitsquare scaled u shifted (current_w*u,current_w*u);
```

This optional loops separate the pieces by thin white lines. By default, this option is turned off.

```
85 if cut_pieces:
    for step_w:= 1 upto w+1:
        draw (step_w*u,u)--(step_w*u,(h+1)*u) withcolor white;
    endfor;
    for step_h:= 1 upto h+1:
        draw (u,step_h*u)--((w+1)*u,step_h*u) withcolor white;
    endfor;
fi;
```

```
86 endfig;
enddef;
```

We output a complete picture first. This picture could be output more efficient, but we saw no real reason for doing so.

```
87 generate_picture (0);
```

The next loop results in upto n pictures, each differing from the previous one. We enter an endless loop that is only exit when we have output enough pictures.

```
88 i := 0;
forever: exitunless i<n;
```

There are 4 different moves: up, down, left and right. We use METAFONT's random generator to choose one.

```
89 x := (uniformdeviate 4);
if x>3:
    if current_w<w: current_w := current_w+1; fi;
elseif x>2:
    if current_w>1: current_w := current_w-1; fi;
elseif x>1:
    if current_h<h: current_h := current_h+1; fi;
else:
    if current_h>1: current_h := current_h-1; fi;
fi;
```

We don't want too much quite legal but unwanted repetition so we keep track of the last one:

```
90 if (current_w<>prev_w) or (current_h<>prev_h):
```

We swap some sub pictures,

```
91 s := r[prev_w][prev_h];
r[prev_w][prev_h] := r[current_w][current_h];
r[current_w][current_h] := s;
```

write down the picture with

```
92 i := i+1;
generate_picture (i);
```

and finally save the last move:

```

93     prev_w := current_w;
       prev_h := current_h;

```

We don't know in advance how many iterations are needed, but at some moment the loop is exit.

```

94     fi;
       endfor;

```

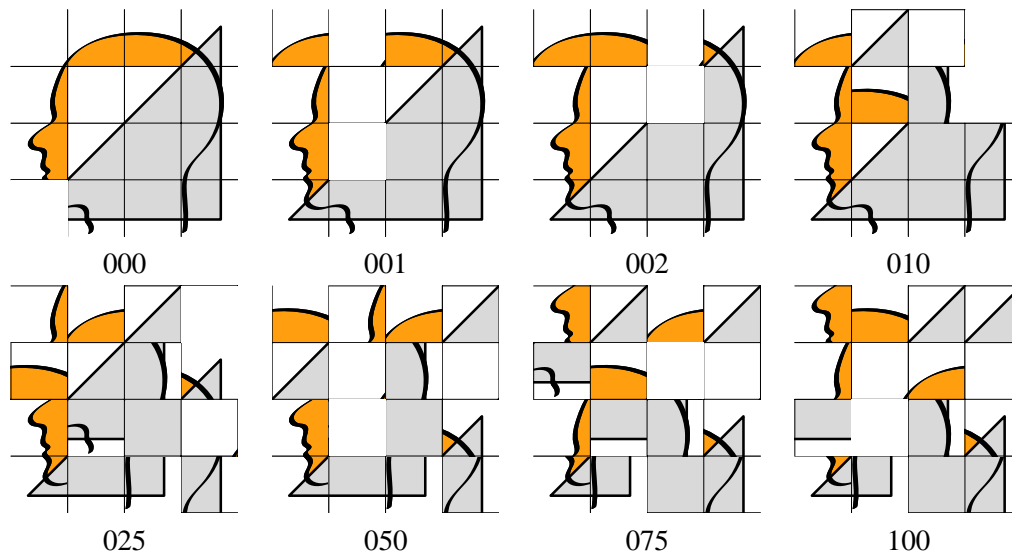


Figure 7: Some simple jigsaws.

It's as simple as that: starting from the solution MetaPost generates the problem. Showing all the results can be very illustrative, but we stick to a subset. The illustration we showed earlier was produced with:

```

95     cut_pieces := true; generate_picture (101);

```

This looks a bit more like a puzzle, doesn't it?

That's it

At the moment we're using MetaPost mainly for generating symbols and logo's of our customers. We plan to use this marvelous program for building libraries of symbols and pictograms and in due time we'll build ourselves a library (macropackage) that enables us to draw all those (simple) math illustrations that we now draw in drawing packages.

Scans and bitmaps in T_EX

Siep Kroonenberg*

n.s.kroonenberg@eco.rug.nl

Abstract

Guidelines are given for working with scans and bitmaps, with the emphasis on PostScript printing from T_EX.

Keywords: Scan, image, bitmap, resolution, halftone, PostScript

Not so many years ago, the only graphics the average computer user got his hands on were graphs from spreadsheets and presentation programs and such, and bitmaps were a last-resort format for these graphics. Nowadays, computing has turned graphical with a vengeance, and many users are in a position to incorporate scanned photographs and other bitmapped pictures into their documents. Time to take a closer look at bitmaps and images.

The emphasis in this paper will be on PostScript printing. When I started using PostScript and Ghostscript, I was delighted at the simple logic of PostScript-based graphics. BM2FONT and emT_EX specials became a bad dream quickly to be forgotten.

Colour models

A bitmap is a grid of pixels which each have colour value. There are bilevel or monochrome bitmaps, grayscale and RGB bitmaps, which require resp. 1, 8 and 24 bits of storage per pixel. There are also palette-based or indexed-colour schemes: the image contains a palette of typically 256 (sometimes fewer, never more) RGB values and the colour of a pixel is represented by a palette index, therefore requiring no more than eight bits.

Tip. Although a scanner may be able to save scans in indexed colour format, it is better to scan in RGB mode and do colour reduction afterwards, e.g. in Photoshop or PaintShop Pro.

Other three-dimensional colour models are HSB (hue/saturation/brightness) and CIE-Lab. CIE-Lab is intended to be device-independent, capable of bridging the differences in colour gamut between different devices (monitors, scanners, colour printers, the human eye). PostScript level 1 does not support the CIE-Lab model.

For print purposes, there is the CMYK colour model, in which colours are defined by percentages of the process colours cyan, magenta, yellow and black. There are many issues involved in RGB-to-CMYK conversion, but when

simply printing to a colour printer, the conversion can usually be left to the printer.

File formats

File formats differ in their coding of pixel information, in compression schemes, in colour models supported and in the additional information they can hold.

TIFF is popular for professional print publishing. It supports all colour models described above and various compression schemes. It may require some trial and error to find out which options work best between two applications.

JPEG supports grayscale and RGB. Its distinguishing feature is its *lossy* compression scheme; in fact, JPEG is both the name of a bitmap format and of this compression scheme. JPEG compression is highly effective even at high quality settings. JPEG compression is designed for (scans of) photographs: such pictures don't compress well with conventional methods but suffer little from JPEG compression. Don't use it for other types of pictures such as graphs, diagrams, logos or screenshots. Also keep in mind that degradation may get worse with successive cycles of editing, saving and reopening: apply JPEG compression once, at the end. JPEG is the preferred choice for publishing photographs on the web.

GIF only supports indexed colour and has a very effective, lossless compression scheme. Its principal use is web pictures. A new alternative is *PNG*, which supports RGB in addition to indexed colour (and nobody claims licencing fees for it). Both GIF and PNG support transparency and interlacing.

For T_EX however, the most important format is probably *EPS* or *Encapsulated PostScript*. The basic idea of EPS is that applications can pass the EPS file to the printer as-is: as long as the originating application can explain its meaning to the printer, the graphic capabilities of the 'host' application, i.e. T_EX, are irrelevant. T_EX can still add wrapper code for positioning, scaling, rotating and cropping.

*This paper is an adaptation of 'Werken met scans and bitmaps' that appeared in our departmental newsletter EZ, 1996/5.

Note that the ‘natural’ dimensions of an EPS bitmap don’t depend on printer resolution.

The EPS format is bulky and has two limitations: it can only be printed to a PostScript printer, and as a rule EPS pictures cannot be read by image-processing software. One important exception is that Photoshop can read back the eps files that it writes.

And then there is a multitude of special-purpose and niche formats such as IMG, BMP, PCX, XMB etc. Use these if you have to.

Resolution

For screen representation, we measure pictures usually in pixels, and picture pixels are preferably mapped one-to-one to screen pixels. For PostScript printing, it makes more sense to talk about physical dimensions on the one hand and resolution on the other, although of course a 1-by-1-inch 600 dpi picture contains the same amount of pixel data as a 2-by-2-inch 300 dpi picture.

Black-and-white bitmapped line art should ideally have printer resolution (after any scaling of course): we want to map image pixels one-to-one to printer dots. Unfortunately this often means a very large number of dots, which may compress to a small file but still have to be expanded inside the printer’s memory. Unlike most or all non-PostScript printers, a PostScript printer will do a lot of interpolating to calculate which dots should be on and which should be off, and might choke on a large monochrome bitmap. Only if it is very clever it might recognize the special case where image dots match up exactly with printer dots. So check with your printshop or service bureau which resolutions they recommend. Below, you see a 300 dpi- and a 600 dpi version of the same picture.



If your black-and-white image is really a halftoned or dithered grayscale picture, make sure that each image pixel maps to an integral number of printer dots, in order to avoid moire effects; see below. Or get hold of the original grayscale image.

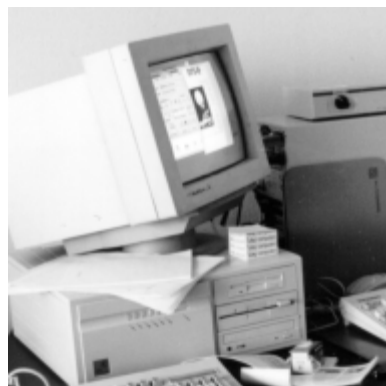
Resolution



75 DPI, 48 KB



150 DPI, 183 KB



300 DPI, 725 KB



For screendumps it is also a good idea to map each image pixel to an integral number of printer dots: screendumps frequently contain dithered areas, and text will also look more even.

Tip. You can set the desired resolution in Photoshop, export the image in EPS format and then place the image in your T_EX document without scaling:
`\includegraphics{picture.eps}`

For photographs, choice of resolution is a different story. Normally you won’t need resolutions exceeding 300 dpi even for high-end output: photographs will be rasterized, and resolution is mainly determined by the screen- or ‘line’ frequency. Pick a resolution of 1.2 to 2 times the line frequency: the higher number if there are sharp transitions, the lower for e.g. cloud pictures:

Device	device dpi	lpi	image dpi
screen	96	n/a	96
laserprinter	300	53	60–100
laserprinter	600	85	100–150
imagesetter	1270	128	150–250
imagesetter	2540	150	200–300

Some line frequencies match up better with the printer resolution than others, and the printer or imagesetter might make adjustments accordingly. Or the printer may support FM (Frequency Modulated) screens or variable dot sizes, which features make the concept of line frequency meaningless.

In the resolution example, the difference in quality between the 150 dpi- and 300 dpi versions is exceedingly insignificant; the difference in file size isn’t.

Gray levels

Higher line frequencies make for sharper pictures. They also mean fewer grays. A few possible screen settings for a 600 dpi printer:

screen	halftone cell	grays
60.6	98	99
70.7	72	73
84.9	50	51 (default)
106.1	32	33
141.4	18	19

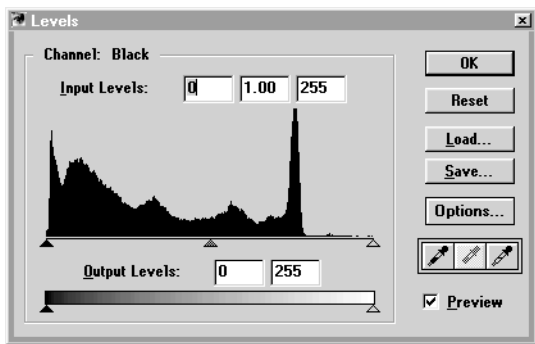
The halftone screens example at the end illustrates some of these settings. These are for 45° screens; for a 0° screen we would have a different series of frequency settings. A full range of 256 grays and a line frequency of 150 lpi requires an imagesetter resolution of at least 2400 dpi. Better halftones are the principal reason for high imagesetter resolutions.

This is not the whole story. Some printers can vary dot size and produce much better grays than the table indicates; a LaserJet 4 printout showed none of the expected stairstepping. On the other hand, the reproduction process imposes limitations of its own on screen frequencies.

You can set a document-wide screen by using Dvips' custom header files. The following code should set a 53 lpi 45° screen:

```
%!
/bop-hook {53 45{dup mul exch dup mul
  add 1. exch sub}setscreen} def
```

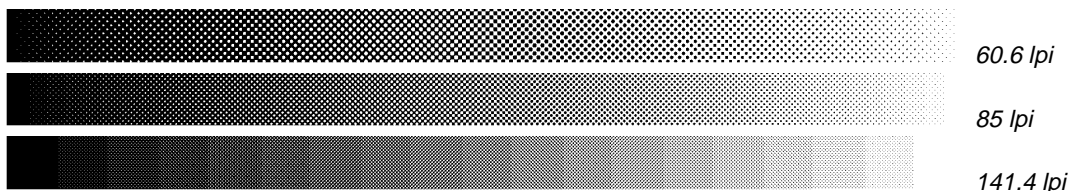
Use this code at your own risk: I have only run a quick test on it and have not used it for production work. Also, Postscript printers and -interpreters may ignore screen settings altogether.



Tuning

Quick-and-easy tonal adjustment: if the tonal range of your picture is e.g. 20–70% black, the contrast will much improve if you stretch this to the full 0–100

Halftone screens



range. The screenshot above displays the Photoshop 3 *image/adjust/levels* dialogue, in which this can be accomplished: move the outer upper two sliders in, ignoring the very thin tail at the right, until they fit snugly around the histogram.

Actually, you might prefer a 5–100 or 5–95 range because patches of pure white or black might look blotchy: slide the lower sliders in to settings of 12 and 243. You can also experiment with the middle of the upper set of sliders to lighten or darken midtones. Make all these adjustments in one visit to this dialog box, in order to minimize quantization errors.

Other programs offer similar facilities although not necessarily as well implemented.

You can also sharpen your scans, but this requires some care, since it easily introduces ugly artefacts.

Printed originals

Scans from printed 'originals' often suffer from fuzziness and moire effects which may be hard to get rid of. Avoid them when you can.

Printing

PostScript and non-PostScript printing each have their own challenges: without PostScript, incorporating graphics is messy at best. With PostScript, it is simple and elegant because you leave the hard work to the printer. If you are careless the printer might choke on the hard work. It helps to restrain yourself with fonts, limit the resolution of images to what is needed and proof your document on a PostScript printer before handing it to someone else.

Tip: if the PostScript 'printer' happens to be GhostScript under Windows, you may get better halftones if you pick the mswinpr2 driver, which lets the default Windows printer driver do the halftoning. You can also try to proof your PostScript with Acrobat Distiller.

References

- A collection of graphics and other file format descriptions can be found at www.wotsit.demon.co.uk.
- For everything related to PostScript, Acrobat and Photoshop, visit Adobe's site: www.adobe.com.
- The Ghostscript site is www.cs.wisc.edu/~ghost.
- A good book on working with scans is *Real World Photoshop* by David Blatner and Bruce Fraser, Peachpit Press, 1996, ISBN 1-56609-169-1.

Using fonts in METAPOST

one way of forcing T_EX to include the right fonts

Hans Hagen

April 1 1997

MetaPost is both an extension and a dialect of METAFONT and is written by John Hobby. This language combines a lot of the strength of METAFONT, T_EX and POSTSCRIPT in one package: a sophisticated graphics language, high quality typography and portability based on outlines.

For some reason, MetaPost needs the public domain DVI to POSTSCRIPT converter DVIPS. This symbiosis originates in the need to include the fonts (glyphs) that MetaPost uses in the POSTSCRIPT file. Driver independancy was one of my prerequisites for using MetaPost, so I decided to build this kind of support myself. Personally I consider driver dependancy a drawback for the dissemination of such a package. This module more or less decouples MetaPost and DVIPS.

The next two commands do the job. They may be called more than once:

```
\UseMetaPostFile      {filenaam}
\UseMetaPostProofFont {fontname}
```

For testing purposes there is:

```
\ShowMetaPostData
```

And for troublesome situations, like independant page processing we've got:

```
\ReUseMetaPostData
```

This module is independant of ConT_EXt, therefore we start with:

```
1 \ifx \undefined \writestatus \input supp-mis.tex \fi
2 \unprotect
```

The process for generating a MetaPost illustration looks more or less like:

```
mptotex filename.mp filename.tex
tex filename.tex
dvitomp filename.dvi filename.mpx
mp filename.mp
```

The file filename.tex contains the T_EX specific commands and ships out a page for every piece of text in the metaposting. These files look like:

```
\shipout\hbox{\smash{\hbox{\hbox{% line 10 examples.mp
$a$}\vrule width1sp}}}}
\shipout\hbox{\smash{\hbox{\hbox{% line 11 examples.mp
$b$}\vrule width1sp}}}}
\end{document}
```

Where the last line takes care of both Plain T_EX and L^AT_EX, and kindly ignores all other formats that redefined \end, which makes it a bit more implementation dependent. By the way, there also seems to be a dependency on \voffset and \hoffset, which should be 0pt while shipping out.

An alternative to the next solution could be a utility that generates a decent prologue file based on the filename.mpx. For the moment we stick to the T_EX based solution.

The T_EX file can be used to determine what fonts and glyphs are needed. We only have to take care of the \shipout and \end. The next command stores the glyphs in a box:

```
\UseMetaPostFile{examples}
```

We reserve a box for this manipulations and append successive calls to `\UseMetaPostFile` to this box. The local redefinition of `\shipout` takes care of the MetaPost ones, the global redefinition is needed later on. We need to reset `\everypar`, otherwise unwanted side effects can occur.

```

3 \newbox\MetaPostData
4 \def\UseMetaPostData#1%
  {\global\setbox\MetaPostData=\vbox
   {\everypar{}
    \unvbox\MetaPostData
    \hbox{#1}}}%
   \global\let\shipout=\MetaPostShipOut}
5 \def\UseMetaPostFile#1%
  {\UseMetaPostData
   {\let\shipout=\relax
    \def\end##1{}
    \input #1\relax}}

```

One may wonder why we don't say `\let\end=\endinput`. It turns out that when we give an `\endinput` in the middle of a sentence, the rest of the line is still processed. This is very handy when writing macros, because these are considered one line. That way `\endinput` can be buried very deep in `\if`'s.

The box is to be shipped out on the first occasion, if possible before the first graphic inclusion, otherwise some drivers still will not be able to produce the right files; one never knows in advance how a driver collects and writes down its fonts. The most secure way of doing this is putting the box somewhere on the page in a white color or scaled to zero. Both mechanism can fail, for instance when we use a background, or when scaling to zero is not supported. By including the box contents, the driver will embed the right glyphs, even when they are out of sight.

We use a brute mechanism and make use of the fact that most viewers and drivers clip the page, due to physical constraints. By putting the glyphs meters above the pagebody, we can be quite sure that they never show up, even on A0 paper format.

We can put the box contents somewhere by hand, but an automatic mechanism is more safe, because that way we can take care of unwanted interference. Putting the glyphs in a box at the top of a page (raised `\maxdimen`) undoubtedly interferes with `\topskip`, so I soon decided to manipulate one of the \TeX primitives that is always used and that could be overloaded without problems. The primitive best suited for this purpose was (of course) `\shipout`. As always, we save the original meaning:

```

6 \let\normalshipout=\shipout

```

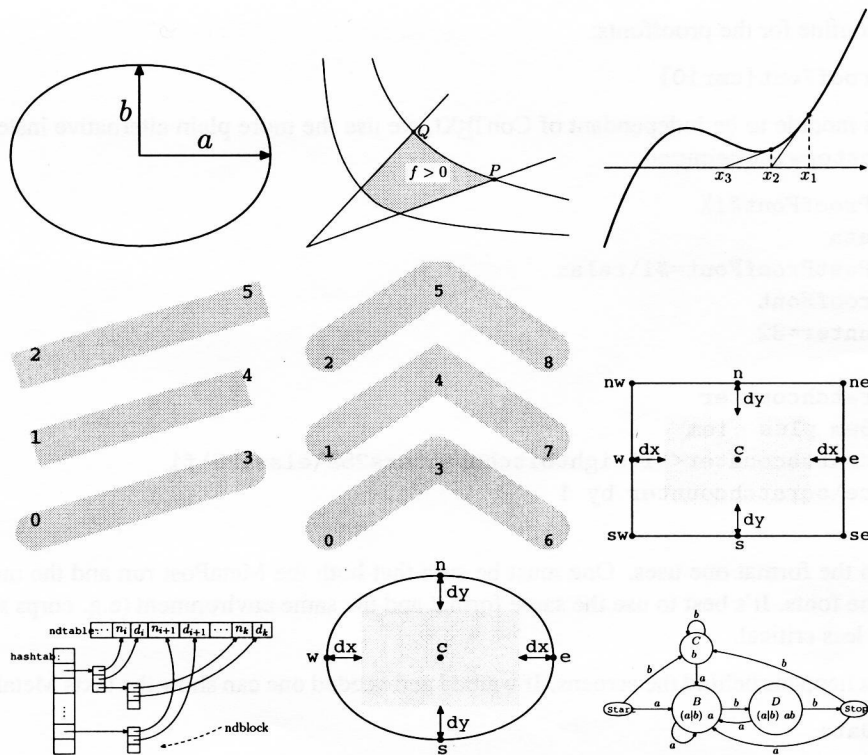
We cannot shipout the box separated from the page, because every `\shipout` generates a page. The next macros do the job.

```

7 \def\DumpMetaPostGlyphs
  {\vbox
   {\wd\MetaPostData=\!!zeropoint
    \ht\MetaPostData=\!!zeropoint
    \dp\MetaPostData=\!!zeropoint
    \kern\maxdimen
    \copy\MetaPostData
    \kern-\maxdimen}}
8 \def\RestoreShipOut%
  {\global\let\shipout=\normalshipout}
9 \def\MetaPostShipOut%
  {\dowithnextbox
   {\normalshipout\vbox
    {\DumpMetaPostGlyphs
     \nointerlineskip
     \box\nextbox}
    \RestoreShipOut}}

```

Now let's prove that things work all right and show the example files that are part of the MetaPost distribution:



In this file we preloaded the right fonts by something like:

```

\bgroup
\switchtocorps      [12pt,cmr,rm] % the 'original' fonts
\UseMetaPostFile    {examples}    % the mp generated file
\UseMetaPostProofFont {cmr10}     % the default prooffont
\egroup
    
```

The first line switches to the default ConTeXt fonts and is package dependant. In real Plain TeX one can omit this line.

When possible, the MetaPost files are to be produced with prologues, which can be accomplished by including the next command in the MetaPost source (the mp file):

```

prologues := 1 ; % this should be default
    
```

If after all these precautions things still go wrong, for instance because the driver produced POSTSCRIPT files on a page by page base, one can use:

```

\ReUseMetaPostData
    
```

After which the 'invisible' box is output at every page. The extra overhead is not that large.

```

10 \def\ReUseMetaPostData%
    {\let\RestoreShipOut=\relax}
    
```

In most cases the amount of extra overhead is small compared to the rest of the data.

MetaPost does not use TeX for typesetting the proofings. This means that we have to load the used proof font, which is cmr10 by default, explicitly. The easiest way of doing this is calling an extra file, in which this font is called:

```

\UseMetaPostFile{mp-proof}
    
```

Such a file looks like:

```

\font\MetaPostProofFont=cmr10

\dostepwiserecurse{48}{127}{1}
  {{\MetaPostProofFont\char\recurselevel\ }}
    
```

or:

```

\font\MetaPostProofFont=cmr10 0 1 2 3 4 5 6 7 8 9
    
```

We provide an extra routine for the proof fonts:

```
\UseMetaPostProofFont{cmr10}
```

Because we want this module to be independent of ConTeXt, we use the more plain alternative instead of the more byte saving alternative `\dostepwiserecurse`.

```
11 \def\UseMetaPostProofFont#1%
    {\UseMetaPostData
     {\font\MetaPostProofFont=#1\relax
      \MetaPostProofFont
      \scratchcounter=32
      \loop
        \char\scratchcounter
        \hskip .5em plus .1em
        \ifnum\scratchcounter<\ifeightbitcharacters255\else126\fi
        \advance\scratchcounter by 1
      \repeat}}
```

Another pitfall lays in the format one uses. One must be sure that both the MetaPost run and the one that generates the document call the same fonts. It's best to use the same format and the same environment (e.g. corps size). Using scalable POSTSCRIPT fonts is less critical.

One may wonder what happens behind the screens. If wanted and needed one can show the texts MetaPost uses by calling:

```
\ShowMetaPostData
```

Because the labels have no height and depth, we use a bit different definition of `\UseMetaPostData`. This time we force a decent linedistance. Because we also want to typeset this data in this text, we also enable linebreaks and correct some spacing. This is how it looks:

```
a| b| f > 0| P| Q| x1| x2| x3| ...| ni| di| ni+1| di+1| nk| dk| :| ndblock
b| - ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; |
M N O P Q R S T U V W X Y Z [ " ] ^ _ ` a b c d e f g h i j k l m n o p q r s t u v w x y z
```

The size of the characters corresponds to the size used during the TeX run needed for the MetaPost job. The vertical spacing is not optimal, but suits its purpose.

The less instructive definitions of both macros complete this module.

```
12 \def\UseMetaPostData#1%
    {\global\setbox\MetaPostData=\vbox
     {\everypar{}
      \unvbox\MetaPostData
      %\baselineskip30pt
      \prevdepth\!!zeropoint
      \ignorespaces#1}%
     \global\let\shipout=\MetaPostShipOut}

13 \def\UseMetaPostFile#1%
    {\UseMetaPostData
     {\def\shipout{ \discretionary{}{}{}}
      \def\end##1{}
      \input #1\relax}}
```

```
14 \def\ShowMetaPostData%
    {\unvbox\MetaPostData
     \vskip15pt}
```

So in order to get the right glyphs in the POSTSCRIPT file we have to:

- say `prologues:=1` in the MetaPost file
- (temporary) activate the fonts used in the graphs
- reuse MetaPost data when needed
- load the proofing font when used

This module will probably be enhanced and/or improved, when I'm past the first experiences with MetaPost. I did consider scanning the POSTSCRIPT file that MetaPost produces. A little bit of scanning and interpreting could do the job quite well, but I wonder if it could be done robust.

15 `\protect`

Stars around I—PostScript straightaway

Kees van der Laan

Abstract

Drawing the outline of stars is discussed. A METAFONT/Post and a PostScript program are included, next to a Columbus' egg PostScript solution. The appendices contain compositions of stars and a few versions of a PostScript operator to calculate the intersection point of two lines determined by four points.

Keywords: EPS, fractal, intersection point 2 lines, Koch island, METAFONT/Post, MFTOEPS, outline, PostScript, ROEX, star, tiling.

1 Introduction

This note arose as a comment, well asked for feedback, on NTG's Graphics course of 1996, where the teacher Jackowski—the \TeX , METAFONT, POSTSCRIPT guru of the Polish GUST user group—extensively used a star as example.¹

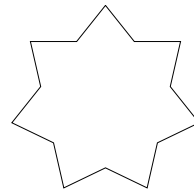
The included small, and hopefully intelligible, codes are in POSTSCRIPT—the portable graphics language, similar to FORTRAN as the portable numerical language—ready to be enriched by other tools such as Adobe Illustrator, to be included as EPS in (La) \TeX scripts, or to be viewed by ghostscript as such.² However, now and then a higher level METAFONT code has been given as well, because of special features of METAFONT, and because Jacko used METAFONT as user language.

In Adobe's POSTSCRIPT tutorial—see their blue book—a star is used as example to illustrate, how to

- translate, rotate and use the CTM³
- execute repeatedly an executable array
- how to fill up the 'contour.'

Jackowski used even-ordered stars to illustrate aspects of METAFONT programming, MFTOEPS—transformation of METAFONT into EPS, and his ROEX—removing overlap utilities.

Example (*Contour of 7-pointed star*)

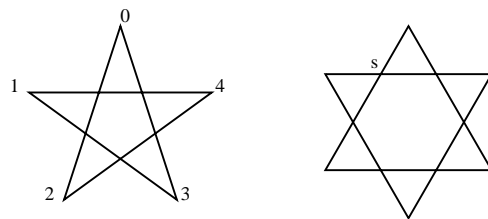


1.1 Notations

What is meant precisely by a star is left to the intuition. The above is a regular, simple and straight odd-ordered star, meaning a symmetric star with an odd number of vertices. Vertices are denoted by z_0, z_1, \dots, z_n . Each z is a pair (x, y) , the conventional way to denote a point in the xy -plane.

For the METAFONT program I did not build a character from it. On my Mac with BlueSky's METAFONT this is not necessary.

Example (*5-pointed and 6-pointed star*)



2 How to draw a star?

A natural approach is to start at a vertex and draw, but ... what are the coordinates of the next turning point?

¹For the material Jacko left behind after the course consult: <http://www.ntg.nl/course18/> or the more elaborate submission from GUST on the CTAN ...

²METAFONT codes need to be converted via MFTOEPS, or MetaPost has to be used in order to yield the PostScript codes.

³CTM stands for Current Transformation Matrix, a central issue in PostScript to relate the so-called *user space* to the *device space* (screen or printer). See the tutorial in Adobe's blue book for how this works.

Adobe circumvents this for the 5-point star by drawing the diagonals of a pentagon—a so-called pentagram—that is

$$z_0 \rightarrow z_2 \rightarrow z_4 \rightarrow z_1 \rightarrow z_3 \rightarrow z_0$$

This is sufficient for filling up the star. A nice way for odd-ordered stars. Their POSTSCRIPT program reads as follows.⁴

```

%!PS-Adobe- 5-pointed star in one-stroke
%%BoundingBox: 0 -50 75 25
/diagonal{72 0 lineto %add to path
  currentpoint translate %move origin
  -144 rotate %rotate dev space
}def
/star5{moveto currentpoint translate
  4{diagonal}repeat closepath
  gsave .5 setgray fill grestore stroke
}def
200 200 star5 %showpage

```

Jackowski added his ROEX functionality in order to retain the structure of the contour, read remove the ‘overlap,’ the inner lines.⁵

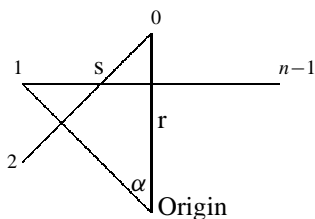
The above 2 stars need an eraser—Jackowski’s ROEX—when one wants just the contour for filling up or coloring purposes. We can do without an eraser in various ways, hang on.

2.1 The point z_s

The coordinates of point z_s obey the following formula,⁶ with $\alpha = 360/n$, n the order,

$$r \cos \alpha \left(-\tan \frac{\alpha}{2}, 1 \right).$$

z_s can be derived from the sketch below.⁷



x_s follows from ‘solving the equation’ for the intersection point z_s of the lines $\overline{z_0 z_2}$ and $\overline{z_1 z_{n-1}}$. This can be done by

METAFONT for sure, but applying a little backside of the envelope geometry, yields

$$x_s = x_2 \frac{y_0 - y_s}{y_0 - y_2}.$$

Drawing the star comes down to drawing the first side $\overline{z_0 z_s z_1}$ and rotate this n times.

3 Star in METAFONT/Post

In METAFONT the computation of the coordinates of the point z_s can be easily specified as can be seen from the following program.

```

%Star contour. Oct 1996. cgl@rc.service.rug.nl.
%Only points z0, z1, z2, z[n-1] are needed.
%Rest follows from symmetry (no ROEX needed.)
n=6; alfa=360/n;
z0=(0,100);
z1=z0 rotated alfa;
z2=z1 rotated alfa;
z[n-1]=z1 reflectedabout (origin, z0);
pair zs;
%equation to be solved by METAFONT
zs=whatever[z0,z2]=whatever[z1,z[n-1]];
%essential side
path p; p=z0--zs--z1;
draw for k=0 upto n-1:
  p rotated (k*alfa)--endfor cycle;
showit;
end

```

The above implies solving 2 equations, although hidden by the declarative specification.⁸ We can do without solving equations. Hang on.

4 Star in PostScript

In POSTSCRIPT the star can be drawn as follows with the use of the formula for the point z_s .⁹

```

%!PS-Adobe- star outline, labels 0, 1, ... n-1
%%BoundingBox: -100 -100 100 100
/n 7 def /r 100 def
/alfa 360 n div def
/cosa alfa cos def
/cos2a cosa dup mul 2 mul 1 sub def
/sina alfa sin def
/sin2a cosa sina mul 2 mul def
/x1 sina r mul neg def %abscis 1st vertex
%inward corner

```

⁴See their blue book p51-53. I used the name diagonal instead of starside, because that is what it is of the pentagon. Jackowski varied these odd-ordered one-stroke star drawings by joins of a spline under tension, yielding pictures which do remind us of flowers instead of stars. Nice. My Malbork window alludes to this in PostScript. Also in ‘Just a little bit of PostScript’ I’ve shown how flowers can be drawn straightaway in PostScript as variant of a line bundle. So polygons, stars and flowers form some sort of continuum.

⁵As he demonstrated during NTG’s Graphics and T_EX course, October 1996. He draws a half-ordered polygon and reflects this in order to draw a star.

⁶In ‘Afterthoughts’ the choice of s is loosened, and restricted only by having equal distance to z_0 and z_1 .

⁷Note that the line $\overline{z_0 z_2}$ is not realistic. So don’t measure the coordinates of z_s . The flavour of the picture is OK and sufficient for supporting a little math.

⁸In PostScript one has to write an operator. Given the operator the specification goes equally simple, modulo some syntactic sugar.

⁹Of course a similar METAFONT program could have been written, equivalent to the PostScript one modula some syntactic sugar.

```

/xs cosa 1 sub cos2a 1 sub div
    sin2a mul r mul neg def
/ys cosa r mul def %also ordinate 1st vertex
%
0 r moveto
n{xs ys lineto x1 ys lineto
    alfa rotate}repeat
stroke %showpage

```

Remarks. For inclusion in a script more than once one can reuse the POSTSCRIPT program parameterized over the order by splitting off the `/n ... def` line and repeat this with the actual value for `n` before each invoke of the constant part. Some sort of ‘two-part’ POSTSCRIPT codes.

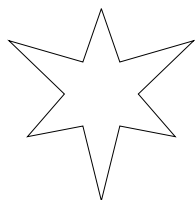
5 Afterthoughts

Looking more closely into the algorithm, biased by the formula for the point s , reveals degeneracy. Instead of the star one gets for

- $n = 3$ a triangle
- $n = 4$ a cross.

A more general—and flexible—approach is to consider the star as a polygon with as side a broken line.¹⁰

We can also variate the location of the corners of the polygon. Because while writing this note it was about X-mas time I have included the following ‘irregular’ star.



The above irregular star is obtained as follows.¹¹

```

%!PS-Adobe- 6-pointed irregular star
%%BoundingBox: -125 -125 125 100
/n 6 def /r 100 def /alfa 360 n div def
/alfah alfa 2 div def
%my choice of rin, make yours
/rin r 2 div alfah cos mul def
0 r moveto
1 1 n{alfah rotate
    0 rin lineto
    alfah rotate
    2 mod 0 eq {0 r lineto}
        {0 r 1.25 mul lineto}ifelse
}for stroke %showpage

```

Do you see how to adjust the above for a ‘pointing’ star, with only one corner stretched?

6 Acknowledgements

Thank you Jackowski for paying us a visit and sharing your experience, next to giving your comments on this

note, especially in suggesting the use of `itransform` for determining the intersection point of two lines. Joseph Romanovsky hinted at other POSTSCRIPT’s features such as the `stack` and `index`. As usual Jos Winnink proofed the paper and lend a helping hand in procrusting the BLUE script into MAPS.

7 Conclusions

Graphics is all about the right strokes and fills. After all no math was needed to draw stars. They can be drawn straightforwardly, as it should be.

For real-life problems the use of Jackowski’s functionalities can be advantageous, however. For his arguments see his lecture ‘A METAFONT-EPS interface,’ EuroTEX 95, Arnhem.

Detailing with stars was nice and familiarized me further with POSTSCRIPT.

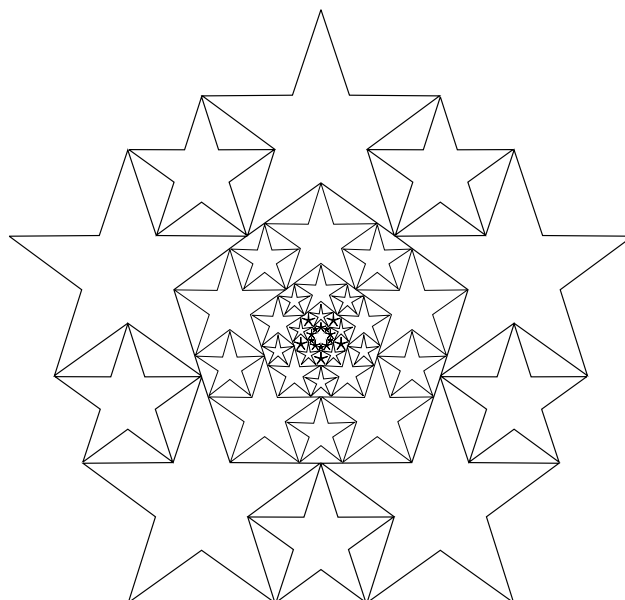
My case rest.

Have fun, and all the best.

Appendix I: Composition of stars

Because we can draw stars so easily it is fun to play a little longer with it. Only the code for the Koch islands is included. I’ll come back on the issues in special notes about tilings and fractals.

Example (*Tiling with stars*)

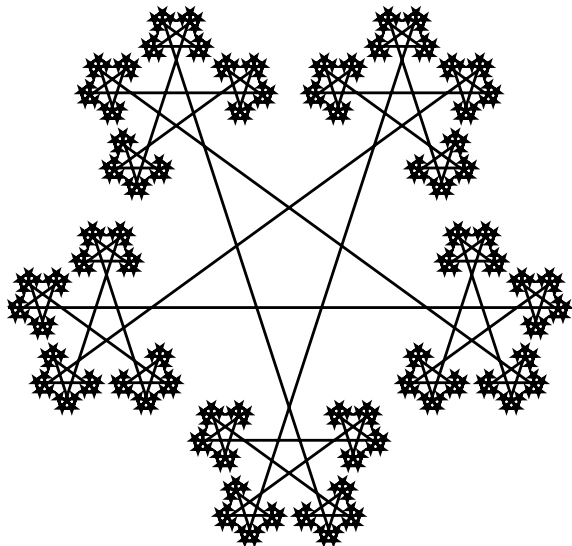


¹⁰Jackowski introduced flexes, that is polygons with splines as side, already at EuroTEX 94.

¹¹The Columbus’ egg solution: a star seen as a polygon with a broken line as side.

Example (*Star fractal*)

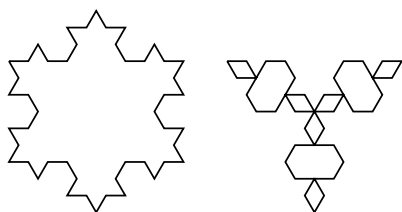
The star fractal picture (and code) is borrowed from Lauwerier's Fractals.¹²

Example (*Koch islands*)

Another class of stars result from wiggly-line fractals as sides of a polygon. A nice example is the so-called Koch island. A Koch fractal—the wiggly-line—results from repeatedly replacing the middle third piece of a line by two in a hat shape.¹³

The star fractal picture (and code) is borrowed from Lauwerier's Fractals.

Below are Koch islands, outbound and inbound, composed from Koch fractals of order 2 as sides of a triangle.



The order 1 outward island is equivalent to the 6-pointed star, discussed in the beginning of this note.

For those who can't wait to play with it the POSTSCRIPT code is appended below. The explanation will be given in the upcoming note about Fractals.¹⁴

¹²Fractals—Meetkundige figuren in eideloze herhaling. Aramith. (It contains codes in BASIC.) When I borrowed material to illustrate turtle graphics in TeX, this fractal was too complicated, because of the drawing at the awkward angles, $144k^\circ$, $k = 0, 1, \dots$. Note that it is just one broken line. The transcription from BASIC into PostScript is a nice exercise for increasing your PostScript fluency.

¹³This scheme of replacing a middle part repeatedly has been used before by Cantor—in his so-called dust fractals—and Sierpiński—in his so-called carpets.

¹⁴What about nesting scaled versions and color these? Bound to be beautiful.

¹⁵The degenerate cases—parallel or co-linear lines—are not taken care of, but can be accounted for, although I don't know how to inform the user. Joseph also suggested to make use of index, to curtail unnecessary roll-s.

```

%!PS-Adobe- Koch wiggly fractal, well islands
%%Creator: Lauwerier; in PS cgl
%%BoundingBox: 0 -80 225 20
/order 2 def
/base 4 def /length 100 def
/linesize length 3 order exp div def
/up base order exp 1 sub def
/koch{0 -60 60 0 %put angles on stack
0 1 up{/k exch def
direction
angle rotate
linesize 0 rlineto
angle neg rotate
}for
}def
/direction{
/angle 0 def
{/angle angle k base mod 2 add
index add def
/k k base idiv def
k 0 eq {exit}if}loop
}def
gsave
0 0 moveto koch
-120 rotate koch
-120 rotate koch
stroke
grestore
125 0 translate
0 0 moveto
-60 rotate koch
120 rotate koch
120 rotate koch
stroke %showpage

```

Remark. The angles can be equally well stored in an array.

Appendix II: The intersection point of two lines

I pondered about a POSTSCRIPT operator for calculating the intersection point of 2 straight lines. Joseph Romanovsky pointed me into the right direction—well, stack-oriented—and provided a tiny POSTSCRIPT program, based on Kramer's rule. I have adapted his code into LU decomposition with partial pivoting.¹⁵ Three variant solutions have been appended.

Algorithm

The process consists of 2 steps

- $z_1, z_2 \rightarrow e, a, b$, ditto for the other points, that is from points to coefficients of the equation of the line determined by those points

$$e = x_1 y_2 - x_2 y_1$$

$$a = y_2 - y_1$$

$$b = x_1 - x_2$$

- solving the equations

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} e \\ f \end{pmatrix}.$$

PostScript code

```

%!PS
/makecoef{%z1 z2 -> e a b
4 copy      %x1 y1 x2 y2 x1 y1 x2 y2
4 -1 roll mul %x1 y1 x2 y2 y1 x2 (y2x1)
3 1 roll mul sub %x1 y1 x2 y2 (y2x1-y1x2)
5 1 roll 3 -1 roll sub
              %(y2x1-y1x2) x1 x2 y2-y1
3 1 roll sub  %(y2x1-y1x2) y2-y1 x1-x2
}def
/solveit{%e a b f c d -> x y,
          %intermediate p is pivot
%Equations: ax + by = e
%           cx + dy = f
%pivot handling %e a b f c d
1 index abs    %e a b f c d |c|
5 index abs    %e a b f c d |c| |a|
gt {6 3 roll} if %exchange 'equations'
%stack: e a b f c d or f c d e a b,
%first is in comments below
exch 4 index   %e a b f d c a
div           %e a b f d p
6 -1 roll dup 6 1 roll 3 1 roll
              %a e b f e d p
4 index exch  %a e b f e d b p
dup 4 1 roll  %a e b f e p d b p
mul sub      %a e b f e p (d-b.p)
4 1 roll mul sub exch div
%a e b (f-e.p)/(d-b.p) = a e b y
dup 5 1 roll mul sub exch div exch
%stack: x y
}def
/intersect {%p1 p2 p3 p4 -> x y
makecoef 7 3 roll
makecoef
solveit
}def
%
%Example
%
100 600 translate
/z1 {10 50} def

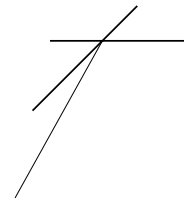
```

```

/z2 {70 110} def
/z3 {100 90} def
/z4 {20 90} def
z1 moveto z2 lineto
z3 moveto z4 lineto
0 0 moveto
z1 z2 z3 z4 intersect lineto
% exchanged equations
%z3 z4 z1 z2 intersect lineto
.1 setlinewidth stroke %showpage

```

with result



Although after all I did not use the above code. I think it is worthwhile nonetheless, and I hope you like it too, if not for the example of how one can do a few calculations in POSTSCRIPT.

Alternative code

After having studied examples in the blue book POSTSCRIPT's parameter mechanism became clear to me, with the advantage of using names instead of muddling around with the stack.¹⁶ Appended is an alternative for /solveit.

```

/solveit{%e a b f c d -> x y
% store stack values by their name
/d exch def /c exch def /f exch def
/b exch def /a exch def /e exch def
a abs c abs lt{exchangeequations}if
/p c a div def
%y=(f-p.e)/(d-p.b), x=(e-b.y)/a
/y f e p mul sub
  d b p mul sub div
def
e b y mul sub a div %x
y                    %y
}def
%
/exchangeequations
{c /c a def /a exch def
 d /d b def /b exch def
 f /f e def /e exch def}def

```

The variables can be made local by using a dictionary. The question arises whether the stack version is sufficient, circumventing variables and pushing an extra dictionary. Make your choice.

¹⁶We can also use arrays in PostScript. For 2-dimensional arrays we must nest 1-dimensional arrays. At the moment I don't know an elegant way of overwriting elements in a nested array. Of course one can store the array stretched and write the algorithm in those terms. However, for this small problem the direct use of the stack looks clear enough.

Using PostScript's itransform

As a comment on this note Jackowski suggested to use itransform. Indeed that is possible.¹⁷ The variant intersect is appended below, without the details of how the POSTSCRIPT matrix should be filled. I have simplified the equations for the occasion by normalizing them, $e = 1$, and $f = 1$.

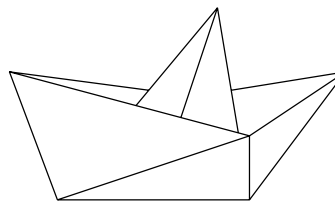
```

%!PS-Adobe- cgl Dec 1996
%Intersection of lines spanned by
%z1, z2 and z3, z4, via itransform
/intersect{%z1 z2 z3 z4 -> x y
  /y4 exch def /x4 exch def
  /y3 exch def /x3 exch def
  /y2 exch def /x2 exch def
  /y1 exch def /x1 exch def
  %makecoef
  1 1 [x1 x2 y1 y2 0 0] itransform
    /b exch def /a exch def
  1 1 [x3 x4 y3 y4 0 0] itransform
    /d exch def /c exch def
  %solveit
  1 1 [a c b d 0 0] itransform
}def
%test
100 600 translate
/z1 {10 50} def
/z2 {70 110} def
/z3 {100 90} def
/z4 {20 90} def
z1 moveto z2 lineto
z3 moveto z4 lineto
0 0 moveto
z1 z2 z3 z4 intersect lineto
stroke showpage

```

Remark. In reproducing the GUST battleship—the EuroTeX'94 logo—the variant with the itransform did not

work???



```

%!PS-Adobe- cgl, Mrt 97
%%BoundingBox: -20 0 112.5 62.5
%/intersect{%see above...
%}def
/s 25 def %scaling parameter
/mean{%p0 p1 on stack -> .5[p0, p1]
  exch 4 -1 roll add .5 mul
  3 1 roll add .5 mul
}def
/p0{0 0}def
/p1{3 s mul 0}def
/p2{4.5 s mul 2 s mul}def
/p3{3 s mul s}def
/p4{-.75 s mul 2 s mul}def
/top{2.5 s mul 3 s mul}def
/p5{p0 top p3 p4 intersect}def
/p6{p0 p1 mean top p3 p4 intersect}def
/p7{top p1 p3 p4 intersect}def
/p8{p2 p5 top p1 intersect}def
/p9{p8 dup 0 exch top p0 intersect}def
gsave
p0 moveto p1 lineto p2 lineto p3 lineto
p0 lineto p1 moveto p3 lineto p4 lineto
p0 lineto
top moveto p5 lineto
top moveto p6 lineto
top moveto p7 lineto
p2 moveto p8 lineto
p4 moveto p9 lineto
stroke grestore %showpage

```

¹⁷Whether to consider this as a feature or a (mis)use that is up to the reader. I guess the reader who is just interested in intersect does not mind what is under the hood.

Stars around II—what a little math can do

Kees van der Laan

Abstract

The problems as treated by Jackowski in his lecture at NTG's 1996 fall meeting: a tangent to a circle, intersection of circles, V-like logo, and (circular) clipping of waves, are provided with alternative solutions, along with straightaway PostScript codes.

Keywords: Analytic geometry, clipping, intersection, linear equations, logo design, METAFONT/Post, MFTOEPS, outline, PostScript, ROEX, tangent.

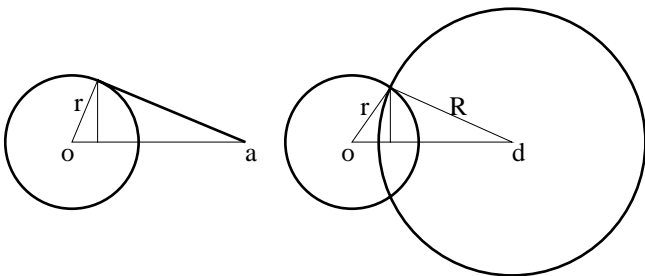
1 Introduction

This note is a summary—and a little more—of Jackowski's lecture on the occasion of him being invited by NTG for the 1996 fall meeting. His lecture comprised the following problems. Drawing

- tangent to a circle
- intersection of two circles
- V-like logo
- clipping by a circle.

I have chosen the origin appropriately and so on, in order to arrive at the simplest representations, unblurred by notational overhead. Furthermore, in the picture I have stressed the circles.

Jackowski emphasized METAFONT's facility to solve (linear) equations. He provided METAFONT codes and transformed these into PostScript. I'll include straightaway PostScript.



2 Tangent to a circle

The problem is: given a circle with radius r , and a point outside the circle at distance a from the center, find the coordinates of the tangent point(s) and draw the picture.¹

For the (left) figure above the tangent points are

$$\frac{r}{a} (r, \pm\sqrt{a^2 - r^2}), \quad r \leq a.$$

The limit situations

$$\begin{aligned} a = r, & \quad (x, y) = (r, 0) \\ a \rightarrow \infty, & \quad (x, y) = (0, \pm r) \end{aligned}$$

can be easily verified.

2.1 PostScript code

The associated PostScript code reads as follows.²

```

%!PS-Adobe- Tangent to a circle
%%BoundingBox: -25 -25 75 25
/Times-Roman findfont 10 scalefont setfont
/r 25 def /a 65 def /rdiva r a div def
/x r rdiva mul def
/y r 1 rdiva dup mul sub sqrt mul def
/angle y x atan def
a 0 moveto 0 0 r angle 360 angle add arc
stroke
%auxiliary lines
a 0 moveto x 0 lineto x y lineto
0 0 lineto x 0 lineto
.1 setlinewidth stroke
%labeling extras
x 2 div 4 sub y 2 div moveto (r) show
-4 -7 moveto (o) show
a -7 moveto (a) show %showpage

```

¹I simplified it by assuming that the origin is at the centre of the circle and that the point a is along the x-axis. Not essential because of the symmetry. Hopefully, confusing non-relevant details have been kept at bay.

²Note how PostScript allows—the operator `arcto`—drawing a tangent to a circle once the tangent point is known. `arcto` is used for drawing rounded corners.

2.2 Jackowski's pedagogical METAFONT code

I have adapted Jacko's codes towards my simplified choice of origin etc. l is the length of the tangent line. Building a character out of it is not so relevant in order to show Jacko's approach—basically equating vectors, declaratively—and therefore I have omitted these aspects, next to the labeling issues.

```
%Tangent to a circle in MF
r:=25; a:=65; l:=a + r;
%Equation in z to be solved by MF
((z-(a,0))rotated-90)/l=z/r;
%Visualize
pickup pencircle scaled 4;
draw fullcircle scaled 2r;
draw z--(a,0);
pickup pencircle scaled 1;
draw z--origin--(a,0);
draw (xpart z,0)--z;
showit
```

3 Intersection of circles

The problem is: given 2 circles with radius r and R , with centres d apart, find the intersection points and draw the picture.

A variant formulation of this problem is: given the length of the sides of a triangle, draw the triangle.³

A little planimetry yields for the intersection points

$$\frac{1}{2d} (d^2 - (R^2 - r^2), \pm \sqrt{(d^2 - (R - r)^2)((R + r)^2 - d^2)})$$

with the restriction $R - r \leq d \leq R + r$.

Limit situations are for $d = R \pm r$ with $(\pm r, 0)$ as intersection points.

3.1 PostScript code

In the PostScript below I've omitted the labeling extras.

```
%!PS-Adobe- Intersection of circles
%%BoundingBox: -25 -50 110 50
/r 25 def /R 50 def /d 60 def
/x d R d div R mul r d div r mul sub sub
  2 div def
/y 1 x r div dup mul sub sqrt r mul def
r 0 moveto 0 0 r 0 360 arc
d R add 0 moveto d 0 R 0 360 arc
stroke
%auxiliary lines
x y moveto x 0 lineto 0 0 lineto
x y lineto d 0 lineto x 0 lineto
.1 setlinewidth stroke %showpage
```

Remark. As can be seen from the illustration the ordinate can be expressed in its abscissa $\pm \sqrt{r^2 - x^2}$, which is computationally simpler, or its variant $\pm r \sqrt{1 - (x/r)^2}$, which

³The pre-computer tool was a pair of compasses. Jackowski called his lecture for that reason I presume 'Virtual ruler, virtual compasses, virtual eraser.'

⁴Once the abscissa is known it's no use to muddle around with vectors. Just a simple Pythagorean subtraction.

is computationally and numerically better, circumventing intermediate overflow. For mathematical purposes the earlier given formula is more suited.

3.2 Jackowski's pedagogical METAFONT code

I have simplified the code for the determination of the abscissa.⁴

```
%Intersection of circles
r:=25; d:=60; R:= 50;
xpart z = .5(d -(R*R - r*r)/d);
ypart z = r + xpart z;
%visualize
pickup pencircle scaled 4;
draw halfcircle rotated -90 scaled 2r;
draw halfcircle rotated 90 scaled 2R
  shifted (d,0);
pickup pencircle scaled 1;
draw z--origin--(d,0)--cycle;
draw (xpart z,0)--z;
showit
```

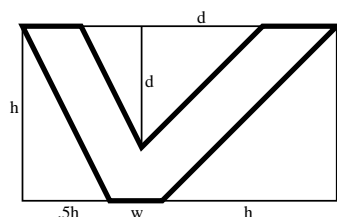
Remarks. In view of METAFONT's limited number range better change in practice

$$(R * R - r * r) / d \rightarrow (R / d * R - r / d * r).$$

METAFONT's `intersectionpoint` can be used, but that is a bit overdone, especially when the analytical solution is at hand. Moreover, the accuracy is low.

4 V-biased logo

The problem is given the height, h , the angles of the legs next to the width of the legs, w , and that the base of the V equals w . Draw the V-biased logo.



The point where the legs meet—determined by d —is implicit. A little backside of the envelope calculation yields

$$d = h - \frac{\sqrt{5} + 2(\sqrt{2} - 1)}{3} w.$$

Remark. If the distance of the incision point to the base should equal w then the formulas are even simpler, and the drawing can better end with a closepath for the base.

4.1 PostScript code

```

%!PS-Adobe- Jacko's V-biased logo
%%BoundingBox: -10 -10 200 100
/h 100 def /w 30 def
/d h 1.021 w mul sub def
0 h moveto
h 2 div 0 lineto
w 0 rlineto
h h rlineto
-1.4142 w mul 0 rlineto
d neg dup rlineto
1.1180 w mul h lineto
closepath
3 setlinewidth stroke
%auxiliary lines
0 0 moveto
0 h lineto
h 1.5 mul w add h lineto
0 h neg rlineto
closepath
1.1180 w mul d 2 div add h moveto
0 d neg rlineto
1 setlinewidth stroke
%labeling extras
/Times-Roman findfont 10 scalefont setfont
-7 h 2 div moveto (h)show
h 5 div -10 moveto (.5h)show
h 2 div w 2 div add 3 sub -10 moveto (w)show
h w add 3 sub -10 moveto (h)show
h 1.5 mul w add 2 add .5 h mul moveto (h)show
1.1180 w mul d 2 div add 2 add
h d 2 div sub moveto (d)show
h h 2 add moveto (d)show %showpage

```

Jacko's METAFONT code solves equations, which is a bit overdone IMHO, with all respect.

5 Clipping by a circle

The problem—a wavy sun—is: create new contours for coloring purposes after clipping the old ones by a circle. Here I made use of simple a priori contours: just broad strokes to conceive the idea.

I color, well... first and then clip. I don't care whether the contours extend the 'window.' This is a different processing order than with Jacko. He clips, rebuilds contours, and then colors or whatever. Definitely more to the point, but

⁵Maybe the problem is induced by METAFONT's lack of a clipping path primitive?

⁶Jacko warned that ROEX is not compatible with MetaPost. The reason is that MetaPost does not make use of pixel values. A year ago or so I experimented with the examples from the ROEX package and could achieve most of the effects if not all in METAFONT alone.

sometimes we can do without. Just be aware of it.



5.1 PostScript code

In PostScript this can be done straightaway as follows.⁵

```

%!PS Tiling Jacko's -s.mf example
%%BoundingBox: -50 -50 50 50
/tile {-2 0 moveto %+/-2 To prevent notch
%any contour or this broad stroke (laziness);
%mind the notch
25 10 25 10 50 0 curveto
75 -10 75 -10 102 0 curveto
} def
/frame{%circle around origin, r=50
50 0 moveto 0 0 50 0 360 arc}def
/dotiling{-50 -50 translate
.9 -.1 .1{setgray tile stroke
0 15 translate}for
}def
gsave
frame clip
1 setgray stroke %white clipped circle
10 setlinewidth %broad strokes
dotiling
grestore %showpage

```

5.2 Jackowski's METAFONT code

The code makes use of `expand_stroke` and `remove_overlap` from his ROEX package, and are beyond my experience as yet.⁶

Jacko is not happy with me calling this an alternative solution. Whatever in the name, I've shown that 80% (or more) of the effect can be done by 20% (or less) of the energy. Moreover, it's just an example.

6 Acknowledgements

Thank you Jackowski for paying us a visit and sharing your experience. As usual Jos Winnink proofed the script and lend a helping hand in procrusting the BLUE script into MAPS submission.

7 Conclusions

It's great that Jackowski showed us how METAFONT can be used to solve and visualize Analytic Geometry exer-

cises, next to sharing his METAFONT, PostScript, and associated tools experience with us.

Our tools become fancier and fancier, but we should not forget to do a few a priori backside of the envelope calculations, whenever feasible. This together with PostScript allows us to include nice graphics at the expense of a little knowledge of PostScript and the merging macros.

My added value is that I've shown that PostScript can be used equally well for visualizing Analytic Geometry exercises, modulo some lumps of syntactic sugar. For me the choice of using PostScript for these problems was necessary for illustrating this note, simply because on my Mac I don't have MFTOEPS running. Next to that I prefer to include pedagogical and intelligible PostScript code as well. They are so concise.

Important is a clear analysis, a minimal algorithmic setup, followed by coding in the language at hand, be it METAFONT, PostScript, or you name it.

I have no access to color printers as yet, and hope that the clipping issues and approaches of the last example have been demonstrated convincingly by grey scales.

A major advantage of Jacko's approach and experiments is that he easily can variate and discover beautiful results. Demonstrating, stimulating creativity. My approach is more constructivistic. I know what I'm after and try to achieve that as simple as possible, in the spirit of Gabo.

My case rest.

Have fun, and all the best.

BachoT_EX'97 — T_EX from inside

Erik Frambach & Kees van der Laan

Abstract

A report of GUST's 5th conference in Bachotek, Poland.

Keywords: Bachotek, BachoT_EX'97, GUST, Poland, Polish T_EX Users Group, conference

GUST has matured as Bogusław Jackowski observed during this meeting, because lectures dealt with general aspects instead of personal (solved) problems, and because significant contributions were done by young GUSTies, such as Piotr Strzelcyk, Joanna Marszałkowska and Marcin Woliński.

Of course we had our 'guitars at night,' but also a banquet à la Polonaise, because of GUST's first lustrum. The bonfire with roasted sausages goes extremely well with a shared Vodka or a beer or two, so ... come, join, contribute and convince yourself next time.¹

More seriously, let us concentrate on the contents of this meeting, although we were as usual hampered by the fact that most presentations were in Polish, except for the few by the foreigners.

1 Conference

Janusz Nowacki was the hero of this conference, not in the least because of his work on transforming the Polish version of the CM fonts into PostScript Type 1.

»AŚŹŹ Łąęśćł«”

Piotr Pianowski reported about his CMYK package. Colour separation of both outline and bitmap EPS graphics is now possible.

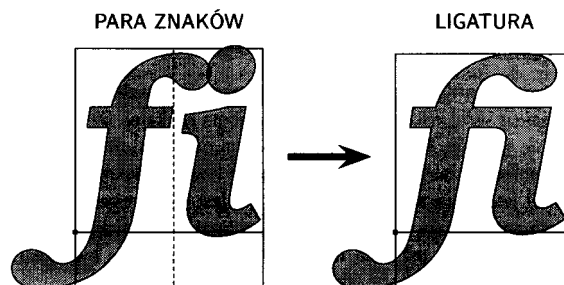
Janusz rehearsed on his Poligraf macro package, emphasizing practical applications.

Andrzej Tomaszewski once again drew attention to typographer's experience, problems and points of view.

Mariusz Olko talked about the benefits of SGML in his tutorial. Aleksander Słomiński reported about the Euromath editor tailored to the SGML- \LaTeX integration.

Jacko, of course, had several contributions. One was about 'PostScript and T_EX' biased by the Polish language.

Włodek Bzyl had an interesting contribution about his adaptation of Knuth's spell-checking approach to Polish.



Rys. 3. Przykład ligatury.

Marek Ryćko showed us nice features of his Bounding Box macros in T_EX. But, ... alas it was in Polish.

Włodek Martin detailed with Polish (T_EXnical) language considerations.

Julita Bolland introduced us to a commercial package for typesetting Chinese. Nice pictograms for sure.

2 From abroad

From the foreigners present Phil Taylor and Kees van der Laan had their contributions in public.

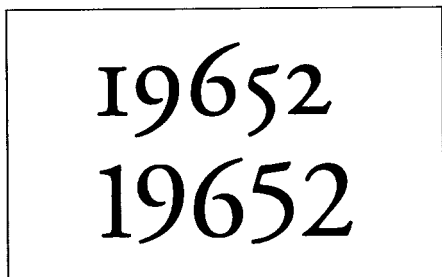
The return of the ligature The return of the ligature

Phil gave a survey of the e-T_EX activities and solicited for comments which seemed a bit misplaced because the proposed features lacked convincing examples for their practical need. He also had his advanced macro writing tutorial, which is really something special.

Apart from some ad hoc problems from the audience he addressed a macro for scanning a PS file for the BoundingBox, characterized by real advanced catcode hackery.

¹Take your shareware—a bottle of Slivowitz for example—with you.

Did you ever think of `\let\oundingBox\fi` within this context? Another issue dealt with on request was real arithmetic in T_EX. His last contribution was about his experience with Adobe's PDF.



Cyfry typu
renesan-
sowego
[nautyczne]
i cyfry «nor-
malne»
kroju pisma,
Adobe
Garamond

Kees reported about 'Just a little bit of PostScript,' and his views on whether we need to know T_EX from the inside or whether insight in T_EX is what we are really after, because we all need to simplify T_EX for personal use and then forget about it.

3 Younger generation

From the younger generation three nice contributions were done.

Piotr Strzelczyk reported about reduction of huge PS files by an order of magnitude. Marcin Woliński, inspired by Kees's FIFO and LIFO, reported about his implementation of stacks and queues in T_EX. Interesting, definitely. Joanna Marszałkowska reported about her translation into Polish of Hobby's Bell reports on MetaPost.

4 Workshops

As usual there were also various workshops for the newcomers. Jacko about PostScript. Staszek about the file server and so on.

5 Conclusions

All-in-all a thorough program. GUST meetings tend to become more and more a family affair. Some were accompanied by their mother, father, or spouse and children. Roughly 100 attendees were present, nearly outnumbering an EuroT_EX.

At home, promises were handled and contacts deepened by email.



Congratulations GUST, indeed you have matured with a very strong kernel of regulars who just come and face the music, next to a growing younger generation, guaranteeing a future for T_EX etc. in Poland.

Een Greep uit de Fontverzameling van T_EX

Ton Biegstraaten*

Samenvatting

De verzameling fonts die beschikbaar is binnen T_EX is zeer omvangrijk. In dit artikel wordt een subjectieve poging gedaan hier iets van te laten zien.

1 Inleiding

Tot ongeveer drie jaar geleden draaiden we hier een zeer oude versie van L^AT_EX (uit 1990). Ontbrekende fonts moesten nog met het handje gemaakt worden, maar het deed wat het moest doen en iets dat werkt moet je niet al te snel veranderen. Toen kwam L^AT_EX2e en dat kon uiteindelijk niet genegeerd worden. Bij het zoeken naar een fatsoenlijk compleet systeem werd N^TE_X gevonden. Dit was alleen geschikt voor Linux, maar na enige moeite was het meeste ook te compileren voor de Sparc.

Na installatie bleken er een geweldige hoeveelheid meta-font files aanwezig te zijn. Bij gebruik ervan werden automatisch .tfm en pk files gegenereerd en dat maakt het toepassen ervan zeer eenvoudig. Echter als je ze zou willen gebruiken moet je ze toch eerst zien, dus heb ik ze afgedrukt.

Tijdens de NTG bijeenkomst in Antwerpen vertelde ik hi-erover bij de lunch (wees gewaarschuwd!) en kreeg het verzoek hier eens wat over op papier te zetten. Door omstandigheden kwam hier niets van. De laatste keer was dit omdat ik hier overschakelde van N^TE_X naar t_EX en alle fonts weer opnieuw kon afdrukken en bekijken, want er was weer veel bijgekomen (overigens ook afgevallen).

Voor de volledigheid, ik ben geen typograaf en heb niet veel verstand van fonts, het kost me i.h.a. al moeite computer modern te onderscheiden van andere fonts, maar ik vind het wél leuk.

2 Wat is t_EX

t_EX is een zeer algemene distributie van T_EX en alles wat erbij hoort, zoals L^AT_EX, amstex etc. In het voorjaar van 1996 is door de T_EX-gemeenschap een cdrom geproduceerd met zeer veel style files, fonts, documentatie en vooral alle binaries voor een groot aantal Unix-achtige machines. T_EX is direct vanaf de CD draaibaar en een volledige installatie op harddisk is zeer eenvoudig. t_EX wordt goed onderhouden en actief ondersteund. Er is een actieve mailing list.

3 Het Afdrukken van Fonttabellen

Het is mogelijk met T_EX van een font de fonttabel inclusief een voorbeeld tekst af te drukken. Start hiertoe T_EX op met de file testfont. T_EX vraagt om een fontnaam en wat je ermee wil (in mijn geval \sample). Door dit nu te doen voor elke .mf file die je tegenkomt en het resultaat (testfont.dvi) te renamen naar de fontnaam met extensie .dvi ontstaan vele .dvi files.

Onder Unix werkt het volgende script:

```
#!/bin/sh
tex testfont <<BLA
$1
\sample
\end
BLA
mv testfont.dvi $1.dvi
```

Om van b.v. cmr10 de testfont pagina te zien moet het volgende commando gegeven worden:

```
script cmr10.
```

Hierbij ga ik er van uit dat het bovenstaande script de naam script heeft. Er zijn intelligentere namen te bedenken overigens.

Door nu alle mfiles van het systeem aan dit script te voeren ontstaan er voor alle fonts voorbeeldfiles in dvi vorm. Er gaat echter ook een boel fout omdat sommige .mf files alleen in combinatie met anderen gebruikt kunnen worden. Er kunnen hierdoor fontpagina's ontstaan die leeg zijn. De achterkant hiervan bleek echter altijd nog als kladpapier bruikbaar.

De hoeveelheid uitvoer kan beperkt worden omdat het niet interessant is alle fontfiles van pointsize 5 tot 25 af te drukken, ook hoeven niet alle varianten van een font te worden afgedrukt. Ik hoef n.l. niet voor iedere familie zowel de roman, de bold, de italic etc versie te zien. Toch blijft er nog veel over.

Resultaat: Een stapel papier van 5 cm.

Dit doorbladerend is het is leuk om te zien wat men allemaal naast reguliere charactersets bedacht heeft aan figur-tjes, symbolen enzo. En altijd zijn er mensen die iets bijzonders willen. Ik geef ze dan de stapel afdrukken. Als

*Faculteit der Technische Wiskunde en Informatica (a.w.w.m.biegstraaten@twi.tudelft.nl)

ze van de schrik bekomen zijn en ze gaan zoeken wordt er vaak wel iets naar wens gevonden. Hierbij scoren vooral de op handschrift gelijkende fonts hoog.

In dit artikel worden alle fonts met de volgende macro afgedrukt:

```
\newcommand{\vb}[2]{\textbf{font: #1}:%
{\font\vbfont=#1 \vbfont #2}}
```

Het eerste argument is de naam van het font, het tweede een voorbeeld string. Deze string bestaat uit gewone letters en cijfers. Voor speciale gevallen gebruik ik i.p.v. een letter of cijfer: `\char'<getal>`, waarbij `<getal>` de octale positie in de door `testfont` gegenereerde caractertabel is.

Sommige fonts zijn wat te klein om een goed beeld te geven. Die heb ik iets vergroot, de vergrotingsfactor staat er bij.

4 Het Aanbrengen van Enige Structuur

In de rest van dit verhaal wil ik de afgedrukte fonts doornemen. Om nu van ieder font alleen maar een voorbeeld op te nemen is naast dat het erg saai is ook weinig informatief. Ik wil proberen enige structuur aan te brengen en hou me daarbij vast aan de volgende punten:

1. Welke min of meer complete fontfamilies bestaan er naast het gebruikelijke computer modern?
2. Welke latijnse alfabetten bestaan er zonder dat er een complete familie varianten van bestaat?
3. Welke fonts met speciale symbolen zijn er?
4. Welke alfabetten uit andere tijden en culturen of zelfs verzonden alfabetten zijn er?
5. Is er nog iets dat dan nog overblijft en toch de moeite waard is?

5 Alternatieven voor de Computer Modern Familie

Van de fonts die nu besproken worden zijn min of meer uitgebreide families aanwezig. Met de geschikte NFSS font definitie files welke i.h.a. in de vorm van packages beschikbaar zijn, zouden ze computer modern kunnen vervangen. Ik laat hier de `dc` (tegenwoordig `ec`) en bijbehorende `tc` fonts buiten beschouwing. Niet omdat ze niet zinvol zouden zijn, integendeel zelfs. Voor veel Europese talen zijn ze een uitkomst. Voor het gebruikelijke werk zien ze er echter (voor een leek) hetzelfde uit als computer modern.

Het is natuurlijk ook interessant te zien hoe een tekst met italic, bold, formules e.d. eruit zou komen te zien in de verschillende fonts en vooral hoe dat voor elkaar te krijgen. Ik wilde me echter hier beperken tot het laten zien van fonts. De L^AT_EX Companion en de bij de fonts behorende documentatie geven voldoende informatie over het gebruik ervan.

Hier volgt van iedere fontfamilie die ik heb een voorbeeld.

Computer Modern Concrete. Zijn gemaakt en gebruikt door Knuth himself tesamen met de euler fonts als wiskundig font. Er is een package `concrete.sty` voor het gebruik binnen L^AT_EX.

font: ccr10:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: eurm10: Voorbeeld 12345

Het computer modern bright font. Gebruik het package `cmbright.sty`

font: cmbr10:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

Malvern font familie. Een sanserif typeface met ook griekse symbolen. De letters en cijfers in de namen van de fonts hebben speciale betekenis welke te vinden is in de documentatie, die nogal uitgebreid is. Je zult daar doorheen moeten om de fonts te kunnen gebruiken. Het geheel maakt een zeer uitgebreide indruk.

font: ma75s10:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

Pandora fonts. Een van de eerste niet computer modern font families. Te gebruiken met het package `pandora.sty`

font: pnr10:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

6 Latijnse Alfabetten zonder Varianten

Het zijn gewone latijnse letters, maar zonder italic, bold etc varianten. Voor speciale toepassingen als aankondigingen prima te gebruiken.

Het meest gevraagd zijn de handschrift letters zoals callig15 die ik wel eens vergroot op een kerstmenu heb gebruikt (package calligra.sty). Opvallend vooral is het font lla14, schoolschrift (van la14) inclusief de lijntjes zoals we hebben leren schrijven (package la.sty).

font: callig15 scaled1440:

*The Quick Brown Fox Jumps Over
Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and
8 but not over 9.*

*Alhoewel in de vorige zin alle letters van
het font voorkomen ontstaat toch geen goed
beeld ervan. Dit komt omdat het font
niet over meerdere regels gebruikt wordt.*

*Door deze tekst er aan toe te voegen
ontstaat een wat langere tekst. Hierdoor
wordt dit hopelijk ondervangen.*

font: cmff10:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: cmfi10:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: la14:

*The Quick Brown Fox Jumps Over
Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8
but not over 9.*

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst

er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: lla14:

*The Quick Brown Fox Jumps Over
Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8
but not over 9.*

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: twcal14:

*The Quick Brown Fox Jumps Over
Lazy Dogs 0, 1, 2, 3, 4, 5, 6,
7 and 8 but not over 9.*

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

Voor de liefhebber zijn er ook een aantal fonts met oude duitse letters. Om ze handig te gebruiken bestaat er een package oldgerm.

font: blackletter:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font

niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: cmfrak scaled1200:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: yfrak scaled1200:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: ygoth scaled1200:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: ysmfrak scaled1200:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: yswab scaled1200:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

Het yinit font is te gebruiken voor de eerste letter van een nieuw hoofdstuk o.i.d. (als je er van houdt tenminste).

font: yinit:



Zelf vind ik het dunhill font altijd wel aardig, misschien omdat ik vroeger pijp heb gerookt.

font: cmdunh10:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

Verder valt xcmr12 op. De doorstreping is niet van mij maar zit in het font.

font: xcmr12:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

Alleen hoofdletters, min of meer gestyleerd.

font: cdr10: ABCDEFG

font: rsfs10: *ABCDEFGHI*

Het o(ptical)c(haracter)r(ecognition) font. Het is geen uitgebreide familie, maar er zijn toch wel verschillende soorten aanwezig waarvan hier een paar opvallende voorbeelden.

font: ocrb10e:

The Quick Brown Fox Jumps Over
Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7
and 8 but not over 9.

Alhoewel in de vorige zin alle
letters van het font voorkomen
ontstaat toch geen goed beeld
ervan. Dit komt omdat het font
niet over meerdere regels ge-
bruikt wordt. Door deze tekst
er aan toe te voegen ontstaat
een wat langere tekst. Hierdoor
wordt dit hopelijk ondervangen.

font: ocrb10f:

The Quick Brown Fox Jumps Over
Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7
and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

Het volgende font heeft dezelfde stijl als teng10, het Tolkien font, maar het is nu leesbaar.

font: engwar:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9:

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan: Dit komt omdat het font niet over meerdere regels gebruikt wordt: Door deze tekst er aan toe te voegen ontstaat een wat langere tekst: Hierdoor wordt dit hopelijk ondervangen:

Je kunt ook pica gebruiken in een paar varianten.

font: cmpica:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: cmpicab:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: cmpicati:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen

ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

Duits schrift, als schoolschrift begin deze eeuw voorgesteld.

font: schwell:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: suet14:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

De volgende fonts zijn er ook nog, zonder commentaar.

font: cmu10:

The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

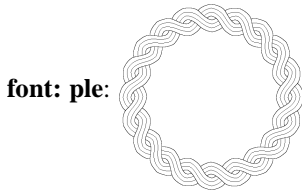
Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

font: cmvt10:

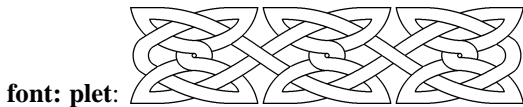
The Quick Brown Fox Jumps Over Lazy Dogs 0, 1, 2, 3, 4, 5, 6, 7 and 8 but not over 9.

Alhoewel in de vorige zin alle letters van het font voorkomen ontstaat toch geen goed beeld ervan. Dit komt omdat het font niet over meerdere regels gebruikt wordt. Door deze tekst er aan toe te voegen ontstaat een wat langere tekst. Hierdoor wordt dit hopelijk ondervangen.

De volgende 3 fonts zijn te vinden onder de croatian taal. Ik rubriceer het maar onder randfiguren. Het eerste font bevat alleen op positie \char'141 een symbool.



De volgende figuur bestaat uit een begin en eind stuk met een tussenstuk. Door het tussenstuk te herhalen kan de figuur verlengd worden. Er is ook een variant met 3 lijntjes i.p.v. 2.



Deze rand is ook een flinke maat groter aanwezig.



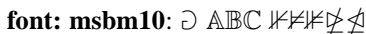
Wie kent ze niet, de Simpsons. Ze horen overigens niet onder het kopje randfiguren al kun je daar over discussiëren



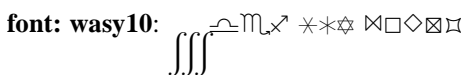
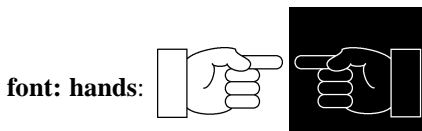
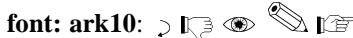
Enige mystieke symbolen, ook te vinden onder de croatian taal.



De American Mathematical Society symbol fonts.

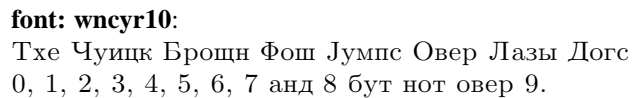
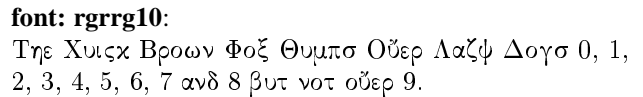
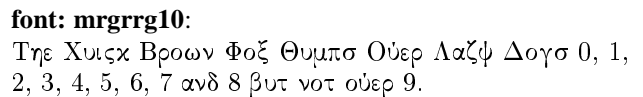
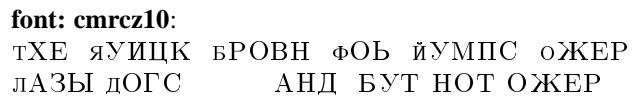
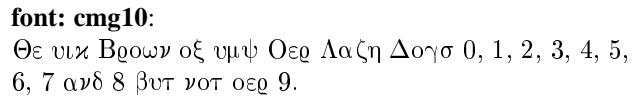


Fonts die voor zich zelf spreken, handjes, pijltjes, krulletjes, wiskundige symbolen, etc

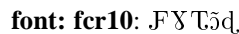


8 Andere Alfabetten

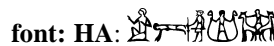
Cyrrillische en griekse tekens zijn ook door een leek vaak nog wel te herkennen. Hiervan een paar voorbeelden.



De fcr, plr, wtkr fonts lijken op het computer modern, ze hebben in positie 128-255 speciale tekens. Van de fcr fonts weet ik dat het afrikaanse tekens zijn, de wtkr fonts zijn turks, de laatste kan ik niet thuis brengen, niet te vinden op CTAN. Het kan heel goed zijn dat de dc en de nieuwe ec fonts ze vervangen. Ik druk wat van de bijzondere tekens af.



Hieroglieven, ik heb er ook een style file voor gezien, hierotex genaamd. Voor de leek zitten er figuurtjes bij die als versiering kunnen dienen. Er zijn veel meer fonts dan ik hier laat zien.



T_EX, BLUes, and Jazz—a search for values

Kees van der Laan

Abstract

How to use T_EX for a database oriented application is shown by the example of a collection of Jazz CDs. The request was to provide (part) of the title of a tune and to let T_EX response with all the CDs which contain that tune. Only a few lines of code was all that was needed to fulfill the purpose. The example might serve as a user-oriented stepping stone for understanding how in BLUe's format T_EX as database engine has been used.

Keywords: BLUe, database, jazz, plain T_EX

Of late I was interested in a way of keeping track of my Jazz collection. Given a tune I wanted to know who had performed it, well ... on the CDs (, records and (vcr) tapes) I own.

A simple application of the database idea, and why not use T_EX for it?¹

1 T_EX as a database engine

On various occasions T_EX has been mentioned as a database tool. Jurriëns has used T_EX as a mail-merge engine. Lenstra and friends launched $\delta\alpha$ T_EX, to handle general data. I myself use T_EX as database ubiquitously. And undoubtedly many others do.

My aim is to share my 'little language,' well a few macro lines, which will achieve the modest request, a very rudimentary database-like 'query.'

At the heart lies BLUe's format `\search` macro, which for this application has been simplified even more, and is called `\logsearch`, because it writes in the log file only.

```
\def\logsearch#1{\immediate
  \write0{Searched for: #1}
  \def\lst##1##2{\loc{#1}{##2}\iffound
    \immediate\write0{\noexpand##1}\fi}
  \input jazz.dat
}
%with auxiliaries
\newcount\cnt \newif\iffound
\def\loc#1#2{\def\locate##1#1##2\end
  {\ifx\empty##2\empty\foundfalse
    \else\foundtrue\fi}\expandafter
  \locate#2.#1\end }
```

`\logsearch` matches its argument with the entries in the data file and it returns the name(s) of the matched entries in the logfile. The reporting is preceded by the search string.

Basically, it comes down to appropriately defining the list element tag, `\lst`, which is the first of 3 elements to characterize each database entry.

```
\lst<name>{<entryproper>}
```

I used as `<entryproper>` the titles of the tunes, each on a line, and as `<name>` I used the main performer, the group-name or any other appropriate identification, with a sequence letter added. To simplify matters further I used lower case characters only and excluded the use of special characters like `&`.

Example (*Search for performers of Summertime*)

The following

```
\newif\iffound \escapechar'055
\logsearch{summertime} \bye
```

yields

```
(jazz.search
Searched for: summertime
(jazz.dat
-blakeyarte
-davisma
-gershwinga
-getzsa
-holidayba
-parkercb
-rossda
-vaughansd
```

¹In fact this application is simple and can be considered as a pedagogical stepping stone to understand the use of T_EX as database engine to handle formats, tools, pictures, addresses, literature items and ilks, as I did in `blue.tex`. As you can see I use few, as few as feasible, structuring tags in the data. (Of course when you provide all parts of an item with a markup tag then you can do anything.) Typing the data remains tedious, alas, and is error-prone. No fancy database-like data entry as yet, alas. But, ... for me an editor and ASCII are sufficient. The big advantage is that I'm *not* dependent on vendors nor versions of database software releases. I can exchange the files easily and reuse them. Crucial is that I have modest wishes.

```
-vaughansg
-vaghansj
)
```

So it is performed by Art Blakey, Miles Davis, on the Gershwin tribute CD, Stan Getz, Billy Holiday, Charlie Parker, Diana Ross, and on various CDs by Sarah Vaughan.

Writing in the logfile is convenient, also when I use it remotely. More than one search can be done in one run.

2 A print of the total collection

Given the names it is easy to spot the entries, be it physically or in the total printout. The latter can be generated by the following job.²

```
\newcount\cnt
\def\{\{\hfil\break}
\def\lst#1{\advance\cnt1
  \\\the\cnt\string#1\quad}
\escapechar'055\obeylines
\input jazz.dat
%Total number of entries at the end
\vfill
```

```
Entries: \the\cnt
\ject\end
```

Each item will be printed and preceded by its sequence number, with the names of the tune on separate lines. The frightening escape character is printed as the friendly hyphen.³

The application is robust against variations in spelling—or typos—because any substring can be specified as search string. The discipline in using comes down to specifying the most relevant part only. Handle data entry carefully and accurately (sigh, what a pleonasm). To verify the relevant parts of the titles is the very least.

3 Conclusion

With a few macro lines a little of the search and report database functionalities have been added to plain T_EX, well ... it can be added to any flavour of T_EX. The problem size is 300 at the moment, the number of my CDs and records in my collection.

Have fun, and all the best.

²With blue.tex it is default in the more economical 2-column.

³Agreed, a very rudimentary realization of a database-like report generation, but for me it is sufficient.

Gegevensverwerking met $\delta\alpha\text{T}_{\text{E}}\text{X}$

Andries Lenstra en Ruud Koning

Samenvatting

De auteurs geven een inleiding in $\delta\alpha\text{T}_{\text{E}}\text{X}$. Dat is een verzameling macro's waarmee gegevens eenvoudig kunnen worden verwerkt in een $\text{T}_{\text{E}}\text{X}$ - of $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -file. Twee voorbeelden illustreren de grote mogelijkheden van deze macro's. Een uitgebreide handleiding en de macro's zelf staan op www.xs4all.nl/~rhkoning.

Keywords: dataverwerking, data-integriteit, $\delta\alpha\text{T}_{\text{E}}\text{X}$.

1 Inleiding

$\text{T}_{\text{E}}\text{X}$ produceert zetsel; wie zich met $\text{T}_{\text{E}}\text{X}$ inlaat, probeert vaak al gauw $\text{T}_{\text{E}}\text{X}$ zodanig in te richten, dat het gewenste zetsel met zo weinig mogelijk intypewerk tot stand komt, al is het alleen maar om de kans op fouten te verkleinen. Het macropakket $\delta\alpha\text{T}_{\text{E}}\text{X}$, dat op deze bladzijden informeel voorgesteld wordt, is bedoeld als hulpmiddel bij genoemde aanpassing van $\text{T}_{\text{E}}\text{X}$ ten behoeve van al die gevallen waarin in het beoogde zetsel naast 'tekst' ook 'gegevens' te onderscheiden zijn.

Gegevens, of 'data', behoren niet overgetypt te worden maar automatisch uit een centrale bron te worden ingelezen. Bovendien moet zo'n centrale bron gemakkelijk in te richten zijn, zonder dat men zich meteen bij dat inrichten voor allerlei keuzen gesteld ziet waar later niet op terug te komen is. De gegevensopslag onder $\delta\alpha\text{T}_{\text{E}}\text{X}$ voldoet ten zeerste aan dat laatste adagium, niet het minst omdat de eerste auteur nooit wijs kon worden uit al bestaande data-base programma's; het is echter niet moeilijk $\delta\alpha\text{T}_{\text{E}}\text{X}$ met bestaande data-bases te laten samenwerken, hoewel nog niet alle code op dat gebied geschreven is. (Momenteel is een windows-programma in ontwikkeling dat gegevens die zijn opgeslagen in een Excel-spreadsheet converteert naar een format dat kan worden verwerkt door $\delta\alpha\text{T}_{\text{E}}\text{X}$.) $\delta\alpha\text{T}_{\text{E}}\text{X}$ haalt de data uit datablokken, en dat zijn (delen van) gewone ascii-files, desnoods deel uitmakend van de onderhavige $\text{T}_{\text{E}}\text{X}$ -file zelf. De bewerkingen die de data vóór het zetten ondergaan, worden verwoord in gewone $\text{T}_{\text{E}}\text{X}$ -code, bijvoorbeeld `\def\Auteur{\bf \Naam}, \Voorletters}` en zijn dus door de meeste gebruikers eenvoudig te beheren.

In de tweede paragraaf schetsen we de ideeën die aan $\delta\alpha\text{T}_{\text{E}}\text{X}$ ten grondslag liggen. Dit gebeurt aan de hand van een voorbeeld dat aan de praktijk is ontleend, een referentie-automaat. Dat voorbeeld krijgt een nadere bespreking in de derde paragraaf. De laatste paragraaf is gewijd aan een tweede praktijkvoorbeeld, de vervaardiging van al het zetsel voor een conferentie. Een nauwkeurige gebruiksaanwijzing van $\delta\alpha\text{T}_{\text{E}}\text{X}$ staat in de Proceedings van de Negende Europese $\text{T}_{\text{E}}\text{X}$ -conferentie in Arnhem, 1995, en op WWW: www.xs4all.nl/~rhkoning. Daar staan

ook de $\delta\alpha\text{T}_{\text{E}}\text{X}$ -macro's, deze tekst en alle code die nodig is voor een werkende implementatie van het eerste voorbeeld. De gebruiksaanwijzing bevat precieze aanwijzingen voor de opzet van het systeem, ook bij zodanige hoeveelheden gegevens dat geheugenbeheer belangrijk wordt ($\delta\alpha\text{T}_{\text{E}}\text{X}$ is in principe tegen zulke situaties opgewassen, want het maakt desgewenst zo weinig control sequences aan, dat het aantal daarvan niet lineair toeneemt met het aantal gegevens in de tekst), en een beschrijving van de 'tools'.

2 Dataverwerking met $\delta\alpha\text{T}_{\text{E}}\text{X}$

Veronderstel dat we een wetenschappelijk artikel aan het schrijven zijn en in dat artikel willen refereren aan andere wetenschappelijke artikelen, de 'referenties'. De gegevens daarvan zullen ooit eenmaal ingetypt moeten worden, dus laten we daar mee beginnen. Dat intypen doen we in een aparte file (bijvoorbeeld `artikel.dat`) of gewoon in de $\text{T}_{\text{E}}\text{X}$ -file waar we mee bezig zijn, op een plaats waar $\text{T}_{\text{E}}\text{X}$ inmiddels $\delta\alpha\text{T}_{\text{E}}\text{X}$ heeft geleerd (`\input datex`). De eerste referentie heet in onze gedachten 'data1', dus als identificerende string die $\delta\alpha\text{T}_{\text{E}}\text{X}$ graag ziet aan het begin van elke regel met gegevens van 'data1', kiezen we de string `data1`. Aan het werk:

```
**data1* \Titel Data with
**data1* \daTeX; \Jaar 1995;
**data1* \Type \Artikel;\EersteAuteur;
**zzzz*
```

en weer ophouden, want er is een kink in de kabel: hoe vermelden we een auteur? Voornaam of voorletters? Of beide? Moet de affiliatie (werkplek) er ook bij? En staan de voorletters bij de eerste auteur achter de achternaam, of gewoon ervoor?

Kortom, het 'veld' `\EersteAuteur` heeft zelf óók velden. Als we de uiteindelijke verschijningsvorm daarvan nog niet willen vastleggen—en dat klopt: we willen die integendeel juist te allen tijde kunnen veranderen—zullen we er nu al over moeten nadenken hoe we die velden als zodanig herkenbaar maken, om ze vervolgens netjes uniform in te typen—twee dingen die we evenmin willen.

Aan al deze tegenzin kan $\delta\text{\TeX}$ drastisch tegemoet komen; dat ontdekken we als we leren wat $\delta\text{\TeX}$ nu al kan beginnen met de luttele gegevens die we het net hebben verstrekt. Als het bovenstaande datablok door $\delta\text{\TeX}$ is geabsorbeerd (c.q. opgezocht, afhankelijk van de gekozen vorm van geheugenbeheer; datablokken zomaar in de tekst worden altijd geabsorbeerd), is van elk veld apart de inhoud toegankelijk:

```
{\bf het artikel \Of *data1* * '\Titel',
 uit \Jaar}
```

resulteert in het zetsel **het artikel 'Data with $\delta\text{\TeX}$ ', uit 1995**, enzovoort (waarbij tussen het tweede en derde sterretje in plaats van een spatie de naam van een data-file ingevuld kan worden als het wenselijk is om $\delta\text{\TeX}$ een aanwijzing te geven omtrent de plaats waar het de string die tussen het eerste en tweede sterretje staat, moet zoeken). Hetzelfde zetsel levert de code

```
\def\Artikel{{\bf het artikel '\Titel',
 uit \Jaar}}%
\Of *data1* * \Artikel
```

en daarvan kan de laatste regel vervangen worden door

```
*data1* *
```

omdat $*data1*$ een afkorting is van

```
{\Of *data1* * \Type}
```

en onze eigen opgave van het $\backslash\text{Type}$ van $data1$ juist $\backslash\text{Artikel}$ luidt. (De gevallen verschillen in de werking van $\backslash\text{Of } *data1* *$; alleen in het tweede geval staat geen $\}$ die deze opheft.)

Nu weten we dus niet alleen hoe we straks gemakkelijk naar artikelen kunnen verwijzen, maar ook hoe we dat probleem met de invoer van auteursgegevens radicaal oplossen: gewoon alle gegevens die we ooit nodig mochten hebben, apart intypen in een datablok:

```
**Andries* \Naam Lenstra;\Vlts A.J.;
**Andries* \Type \Auteur;
**zzzz*
```

en het in het begin gestaakte intypen zo hervatten:

```
**data1* ... \EersteAuteur *Andries* *;
**zzzz*
```

dus als veldinhoud geen expliciete gegevens, maar een verwijzing daarnaar! (Men zou zulke verwijzingen 'meta-data' kunnen noemen; we zagen al een eenvoudiger maar niet minder praktisch voorbeeld van meta-data toen in het veld $\backslash\text{Type}$ een control sequence werd neergezet.) Met de werkplek gaat het desgewenst net zo:

```
**Andries* \Naam Lenstra;\Vlts A.J.;
**Andries* \Type \Auteur;
**Andries* \Werkplek *UvA* *;
**UvA* \Type \Affiliatie;\Adres Plantage
**UvA*Muidergracht 24;\Pcode 1018 TV;
**UvA* \Plaats Amsterdam; \Naam W.I.N.S.
**UvA*Universiteit van Amsterdam;
**zzzz*
```

Op de mogelijkheden die $*\langle\text{identificerende string}\rangle*$ als veldinhoud biedt, komen we straks terug; eerst geven we de data-typist nog enige aanwijzingen. Het prettige besef, geen woord teveel in te hoeven typen, zal diens ijver ongetwijfeld verdubbelen.

Alle regels die met dezelfde $**\langle\text{identificerende string}\rangle*$ beginnen, horen bijeen te staan; als dat nog niet het geval is, is het met vele sorteer-programma's tot stand te brengen (en als dat wel het geval is, hoeft alleen de eerste $**\langle\text{string}\rangle*$ te blijven staan). De regel met $**zzzz*$ is precies daar vereist waar een datablok ophoudt, dus waar de data-file eindigt c.q. de tekst verder gaat, of waar het tijd is voor het poneren of opheffen van verstekvelden. Een verstekveld is van toepassing op alles wat volgt, tenzij het ter plaatse terzijde wordt geschoven, namelijk door een expliciet gegeven veld van dezelfde naam, of tezamen met alle andere verstekvelden wordt opgeheven door het commando $\backslash\text{NoDefaults}$. Als er vele auteurs achter elkaar moeten worden ingevoerd, hoeft er dus maar één keer te worden meegedeeld dat het gaat om het $\backslash\text{Type } \backslash\text{Auteur}$; die mededeling geschiedt door de regel

```
\Default \Type \Auteur;
```

Met behulp van dit mechanisme is het extra werk dat de compositie van een identificerende string met zich meebrengt, tot een minimum terug te brengen; in het geval van $\backslash\text{Type } \backslash\text{Auteur}$ ligt het bijvoorbeeld voor de hand om die string gelijk te doen zijn aan de achternaam, en dat—of eigenlijk het omgekeerde—gaat zo:

```
\Default \Naam \Ident;
```

omdat de control sequence $\backslash\text{Ident}$ steeds de identificerende string geeft. De volgende datablokken illustreren het bovenstaande.

```
\NoDefaults
\Default \Type \Auteur;
\Default \Naam \Ident;
**Andries* \Naam Lenstra; \Vlts A.J.;
**Andries* \Werkplek *UvA* *;
**Andries* \Voornaam \Ident;
**Koning* \Vlts R.H.;\Voornaam Ruud;
**Koning* \Werkplek *VU* *;
**UvA* \Type \Affiliatie;\Adres Plantage
**UvA*Muidergracht 24;\Pcode 1018 TV;
**UvA* \Plaats Amsterdam; \Naam W.I.N.S.
**UvA*Universiteit van Amsterdam;
**VU* \Type \Affiliatie;
**VU* \Naam Vakgroep Econometrie
**VU*Vrije Universiteit;
**VU* \Adres De Boelelaan 1105;
**VU*\Pcode 1081 HV;\Plaats Amsterdam;
**zzzz*
\NoDefaults
\Default \Type \Artikel;
**data1* \Titel Data with \daTeX;
**data1* \Jaar 1995;
**data1* \EersteAuteur *Andries* *;
**zzzz*
```

Om zich voor te stellen wat er zoal binnen bereik komt als `*<een identificerende string>*` in een veld staat, is maar weinig fantasie vereist. Met de juiste definities van de `\Typen` in kwestie zijn velden van velden van ... velden gemakkelijk toegankelijk:

```
\def\Artikel{\bf\EersteAuteur}%
\def\Auteur{\Werkplek}%
\def\Affiliatie{\Plaats}%
*data1* *
```

geeft als zetsel de plaats van de werkplek van de eerste auteur van het artikel `data1`, dus **Amsterdam**. Zo'n faciliteit komt goed van pas in combinatie met het $\delta\text{\TeX}$ -commando `\Filter`:

```
\Filter artikel \Type \Artikel
```

bekijkt de file `artikel.dat` van het begin tot het einde en zet de `\Artikelen` die het tegenkomt, neer in de huidige file volgens de implementatie van `\Artikel` die op dat moment van kracht is. (Dit is dus het gereedschap om een 'mail merge' mee te bewerkstelligen.) In de veronderstelling dat alle gegevens inmiddels netjes zijn ingetypt en opgeborgen in aparte data-files, geven we een van de manieren om alléén díe artikelen te selecteren waarvan de eerste auteur zijn werkplek in Amsterdam heeft, onder gebruikmaking van een paar nog niet genoemde stukken gereedschap.

```
\input datex
%
% onderwijst \TeX \daTeX
%
\TheDataFiles
auteurs (a)
\InSearchOrder
%
% de file auteurs.dat, waarin de
% \Auteurs en de \Affiliaties staan,
% wordt geabsorbeerd (a)
%
% (bij zeer grote data-files moet
% en kan dit anders; zie de gebruiks-
% aanwijzing)
%
\def\Affiliatie{\WrapIn\pLaats\Plaats}%
%
% dit komt neer op:
% \gdef\pLaats{inhoud van \Plaats>}
%
\def\Artikel{%
\def\pLaats{niks}%
%
% geeft \pLaats eerst een verstekwaarde
%
\def\Auteur{%
\IfField\Werkplek\Exists
\Werkplek
\Fi
}%
%
% (een \Auteur hoeft geen \Werkplek
% te hebben, maar elke \Affiliatie
```

```
% heeft een \Plaats)
%
\EersteAuteur
%
% definieert eventueel \pLaats opnieuw
%
\IfCs\pLaats\IsDefinedAs{Amsterdam}%
\def\Auteur{\Vlts\ \Naam}%
'Titel' van \EersteAuteur,
uit \Jaar\par
%
% anders: niets doen
%
\Fi
}%
%
\Filter artikel \Type \Artikel
```

Dezelfde werking zou ook te verwezenlijken zijn zonder `\pLaats` te definiëren, met behulp van het commando

```
\IfExistingField\LooksLike
\Plaats Amsterdam;%
...
\Fi
```

In beide gevallen had het niets uitgemaakt als de `\Plaats` óók weer een verwijzing was geweest, dus `\Plaats *Amsterdam* *`; in plaats van `\Plaats Amsterdam;`, behalve dat de voorwaarde dan had moeten luiden

```
\IfCs\pLaats\IsDefinedAs{*Amsterdam* *}%
```

respectievelijk

```
\IfExistingField\LooksLike
\Plaats *Amsterdam* *;%
```

We besluiten deze inleiding met een geruststelling voor de \TeX pert en een kleine opgave aan hetzelfde adres: na `\Of *data1* *` horen natuurlijk alle control sequences die veldnamen voorstellen, de inhoud te geven die bij `data1` hoort. Stel dat er een tweede artikel, `data2`, ingevoerd is, maar zónder `\Jaar`.

Na `\Of *data1* *` is `\Jaar` kennelijk zo gedefinieerd dat \TeX uit `{\bf\Jaar}` tenslotte het zetsel **1995** produceert, maar bij de tweede control sequence `\Jaar` in

```
\Of *data1* * \Titel, uit {\bf \Jaar}, en
\Of *data2* * \Titel, uit {\bf \Jaar}
```

hoort \TeX níet **1995** te geven—en dat doet het gelukkig ook niet; er komt een foutmelding. Wat weerhoudt \TeX ? Bedenk dat het níet eerst, zodra het `\Of *data2* *` tegenkomt, een vaste lijst veldnamen op een verstekwaarde mag zetten, want zo'n lijst is er niet. In de oerversie van $\delta\text{\TeX}$ deed \TeX dat wel, maar het bijhouden van een lijst veldnamen is een onding voor de data-typist.

3 Toepassing 1: een referentie-automaat

Hierboven werd ter inleiding in de werking van $\delta\alpha\text{T}\text{E}\text{X}$ een referentie-automaat opgezet. Het moge vanzelf spreken dat de manier waarop dat gebeurde, geenszins onder alle omstandigheden de beste hoeft te zijn en dat $\delta\alpha\text{T}\text{E}\text{X}$ vele andere mogelijkheden ondersteunt. In deze paragraaf wijzen we op een voor de hand liggende vereenvoudiging, ontwerpen we nuttige stukjes $\delta\alpha\text{T}\text{E}\text{X}$ -gereedschap (lang leve TEX als programmeertaal!) en lichten we een paar onderdelen toe van de code die op het net staat. Tot slot stuiten we op een nieuwe faciliteit van $\delta\alpha\text{T}\text{E}\text{X}$ door ons voor te stellen dat er heel veel referenties zijn, zodat we zo zuinig mogelijk met het werkgeheugen van het rekentuig om moeten gaan.

De voor de hand liggende vereenvoudiging schuilt natuurlijk in een minder drastische aanpak van de velden van velden. Als op voorhand vast staat dat affiliaties en voornamen nooit vermelding vinden, is het heel goed mogelijk om van het `\Type \Auteur` af te zien en de `\Auteurs` op te geven als, bijvoorbeeld,

```
**data1* \Auteurs Lenstra, A.J.,
**data1*<andere auteurs>, Koning, R.H.;
```

TEX zoekt dan zelf uit hoeveel auteurs er zijn, en bij welk rangnummer de gegeven achternamen en voorletters horen. Veel eenvoudiger te implementeren is het om steeds maar één auteur tegelijk op te geven, net als in paragraaf 2. Bij vier auteurs zou dat zó kunnen:

```
**data1* \EersteAuteur Lenstra, A.J.;
**data1* \EersteTussenauteur ..., ...;
**data1* \TweedeTussenauteur ..., ...;
**data1* \LaatsteAuteur Koning, R.H.;
```

Als er altijd na de komma vóór de voorletters een spatie staat, geven de volgende `\NaamEersteAuteur` en `\VltsEersteAuteur` wat ze beloven:

```
\let\ea\expandafter
%
% (doet \daTeX zelf al)
%
\def\VoorDeKomma#1, #2+{#1}%
\def\NaDeKomma#1, #2+{#2}%
\def\NaamEersteAuteur{%
  \WrapIn\eaEersteAuteur\EersteAuteur
}% of:
% \edef\eaEersteAuteur{\EersteAuteur}%
  \ea\VoorDeKomma\eaEersteAuteur+}%
\def\VltsEersteAuteur{%
  \WrapIn\eaEersteAuteur\EersteAuteur
}% of:
% \edef\eaEersteAuteur{\EersteAuteur}%
  \ea\NaDeKomma\eaEersteAuteur+}%
```

De code die op het net staat, hanteert wél het `\Type \Auteur`, zoals uiteengezet in de tweede paragraaf, maar niet het `\Type \Affiliatie`, omdat de behoefte aan de vermelding van een `\Werkplek` zich nog niet heeft voorgedaan. Dat veld toevoegen kan altijd nog.

Met een apart `\Type \Auteur`, en elke auteursvermelding in de vorm `*Andries* *`, hoeven we, om te voldoen

aan de vaak gestelde eis dat alleen van de eerste auteur de `\Naam` aan de `\Vlts` voorafgaat, in de definitie van `\Artikel` alleen de definitie van `\Auteur` te veranderen:

```
\def\Artikel{...
  \def\Auteur{\Kap\Naam, \Vlts}%
  \EersteAuteur
  \def\Auteur{\Vlts{} \Naam}%
  <eventuele overige auteurs>
  ...
}%
```

`\Kap` zorgt ervoor dat de lezer gruwelen als **het artikel van de Boer [1993]** bespaard blijven, door het erop volgende token helemaal te expanderen en de eerste letter van het resultaat kapitaal te zetten, dus als hoofdletter. Een werkende implementatie is:

```
\let\ea\expandafter
\def\Kap#1{\edef\ExpToken{#1}%
  \ea\Hap\ExpToken}%
\def\Hap#1{\uppercase{#1}}%
```

en met `\def\dH{de heer}` produceert

```
**de Boer* \Naam \Ident;
**de Boer* \Type \Auteur;\Titel \dH;
**zzzz*
\def\Auteur{{\bf\Kap\Titel\ \Kap\Naam}}%
*de Boer* *
```

het zetsel **De heer De Boer**; ook het systeem dat in de vierde paragraaf ter sprake komt, kan het niet stellen zonder de diensten van een commando als `\Kap`.

De laatste opmerking die we over de code op het net willen maken, betreft de verschillende manieren waarop $\delta\alpha\text{T}\text{E}\text{X}$ weet te achterhalen of het de eerste keer is dat naar een bepaalde referentie verwezen wordt. ($\delta\alpha\text{T}\text{E}\text{X}$ wil namelijk in dat geval álle auteurs vermelden, en niet volstaan met de eerste auteur ‘et al.’.) Kort en snel is de code die berust op de eigenaardigheid van TEX , waar onlangs weer door Paul Taylor op is gewezen, dat een control sequence die gevormd wordt met behulp van `\csname` en `\endcsname` equivalent is aan `\relax` zolang hij nog niet met `\def` een betekenis heeft gekregen. `\Ident` leent zich bij uitstek voor de vorming van zo’n control sequence; als deze meteen na de toets die hem gelijk bevindt aan `\relax`, ge\gdefinieerd wordt als willekeurig wat (`\global` omdat alles zich in een groep afspeelt), zal hij alle volgende keren ongelijk zijn aan `\relax`.

Handig, maar wel een belasting voor het werkgeheugen; elke eerste verwijzing stopt daar zo immers een nieuwe control sequence in. Als het werkgeheugen voor die belasting te klein is, moet het schijfgeheugen te hulp komen: $\delta\alpha\text{T}\text{E}\text{X}$ dient dan de identificerende strings die het ontmoet, naar een file weg te schrijven, en die file te inspecteren als het wil weten of het een bepaalde string al eerder is tegengekomen. Dit is met weinig moeite voor elkaar te krijgen; met iets meer moeite is zelfs te bereiken—en

daar is natuurlijk geen geheugengebrek voor nodig—dat de data-file van `\Artikelen` met de nieuwe verwijzingsgegevens wordt ververst. In de uiteindelijke lijst referenties kunnen dan bij elk artikel de nummers staan van de bladzijden waarop naar dat artikel wordt verwezen, waardoor die lijst veel bruikbaar is dan men meestal aantreft.

Het enige wat hiervoor nodig is, is ervoor te zorgen dat het wegschrijven, waartoe de opdracht in `\Artikel` staat, gebeurt in het format van een regel van een $\delta\text{T}\text{E}\text{X}$ data-file, met na de identificerende string nog een veld waarvan de inhoud het paginanummer van de verwijzing karakteriseert, en de naam het rangnummer daarvan plus het artikel waarin verwezen wordt. Stel dat dat `data4` heet en dat daarin op bladzijde 717 voor de negenentwintigste maal `*data1* *` staat, dus verwezen wordt naar `data1`, dan zou de op die plaats weggeschreven regel er zo uit kunnen zien:

```
**data1* \VerwijzingNrXXIXinDataIV p. 717;
```

De inspectie op strings die al geweest zijn, is eenvoudig te verwezenlijken door de wegschrijf-file te `\Filteren` in de staat waarin die zich op dat moment bevindt. (Als $\delta\text{T}\text{E}\text{X}$ gaat opzoeken in plaats van filteren, dan denkt het dat het klaar is zo gauw het de eerste regel gevonden heeft, terwijl het met `\Filter` de hele file doorloopt.) Om de data-file te verversen hoeven we alleen maar de regels van de volledige wegschrijf-file op de juiste plaatsen in de oorspronkelijke file `artikel.dat` in te voegen. `\Filteren` van het resultaat levert tenslotte de lijst referenties volgens de beloofde specificatie.

4 Toepassing 2: een programmaboekje

De Vereniging voor Statistiek en Operationele Research (VVS) organiseert jaarlijks de zogenaamde Statistische Dag. Op deze dag worden 's ochtends twee hoofdlezingen gegeven en 's middags verschillende presentaties in parallelsessies. Het programma wordt gedrukt in een programmaboekje, dat naast het programma ook samenvattingen van de voordrachten en enige advertenties bevat. Voor de samensteller van het boekje kan de uiteindelijke productie ontaarden in buitengewoon veel en onaangenaam werk: hij (of zij) moet ongeveer twintig sprekers benaderen, van alle sprekers een titel van de lezing opnemen in het programma, en twintig samenvattingen laten afdrukken in het boekje. Gegevensintegriteit is in dit geval buitengewoon belangrijk: het is pijnlijk als een spreker aangeschreven wordt als 'Ruud Koning', in het programma vermeld staat als 'drs Ruud H. Koning' en volgens de samenvatting 'dr. R.H. Koning' heet. Verder staat het slordig als de ene spreker met een voornaam wordt aangeduid en een andere spreker met initialen. Daarnaast zijn fouten in de titelaanduiding snel gemaakt (schrijven we 'prof. dr.', 'prof.dr' of 'prof.dr.'). Veel van dit soort problemen kunnen worden voorkomen door $\delta\text{T}\text{E}\text{X}$ te gebruiken. Het schrijven

van brieven en rekeningen, ook aan de adverteerders, en het maken van een programmaboekje kunnen hiermee in hoge mate worden geautomatiseerd, en dat betekent dat de organisatie zich niet bezig hoeft te houden met typewerk, maar zich kan richten op het organiseren van de dag zelf. Als het verder niet veel werk is om een mailing te versuren aan de sprekers, is er geen reden om de samenvatting en personalia (die vaak niet rechtstreeks van de spreker afkomstig zijn en soms handgeschreven terechtkomen bij de organisator) niet door de sprekers te laten corrigeren, of om uitgenodigde sprekers na de Statistische Dag niet schriftelijk te bedanken voor hun voordracht.

In deze toepassing worden twee bestanden met gegevens gebruikt: een bestand met lezingen (`lezing.dat`) en een bestand met personen (`spreker.dat`). Een lezing heeft de volgende datavelden: `\titel` met de titel, `\spreker` met de spreker en `\samenvatting` met de filenaam waarin de samenvatting staat. Een voorbeeld is¹

```
**hfd1* \titel Vijftig jaar VVS: nog%
**hfd1* eenmaal de magere jongelieden;
**hfd1* \spreker *zwet* *;
**hfd1* \samenvatting vzwet.tex;
**hfd1* \Type\lezing;
```

In het bestand `spreker.dat` komen de volgende regels voor:

```
**zwet* \naam van Zwet;\titels \profdr;
**zwet* \vlts W.R.;\Type \persoon;
**zwet* \vakgroep Vakgroep Mathematische%
**zwet* Statistiek;\pcode 2333 CA;
**zwet* \straat Niels Bohrweg 1;
**zwet* \plaats Leiden;
```

Zoals eerder betoogd is het niet noodzakelijk om de verschillende gegevenstypen (lezingen en personen) in verschillende bestanden op te slaan. Met behulp van deze gegevens zijn we in staat om bepaalde elementen uit het programmaboekje te maken. Allereerst kondigen we de lezingen aan in het programma zelf. Later moet de samenvatting van de lezing worden afgedrukt. De aankondiging in het programma bestaat uit de titel en de spreker.

De volgende macro's zorgen ervoor dat een lezing wordt afgedrukt:

```
\newcommand{\profdr}{prof. dr. }
\newcommand{\Spreker}{}
\newcommand{\Titels}{}
\newcommand{\Vlts}{\Kap}
\newcommand{\Titel}{}

\newcommand{\persoon}{%
\renewcommand{\Titels}{}%
\IfField\titels\Exists
\renewcommand{\Titels}{\titels}%
\Fi
\renewcommand{\Vlts}{\Kap}%
\IfField\vlts\Exists
\renewcommand{\Vlts}{\vlts\ }%
}
```

¹De voorbeelden zijn gebaseerd op het programma van het jubileumcongres van de VVS op 5 en 6 april 1995 en zijn enigszins vereenvoudigd.

```

\Fi
\Titels\Vlts\naam
}

\newcommand{\lezing}{%
\renewcommand{\Titel}{titel volgt}%
\IfField\Titel\Exists
\renewcommand{\Titel}{'\titel'}%
\Fi
\renewcommand{\Spreker}{spreker onbekend}%
\IfField\spreker\Exists
\renewcommand{\Spreker}{\spreker}%
\Fi
{\bf \Titel, \Spreker}%
}

```

Als de tekst van het programmaboekje `*hfd1* *` bevat, resulteert dit in het zetsel **‘Vijftig jaar VVS: nog eenmaal de magere jongelieden’, prof. dr. W.R. van Zwet**. Als we kijken naar de definitie van de macro `\lezing` is duidelijk wat er gebeurt. Elke macro van het type `\lezing` voert uiteindelijk tot de expansie van de macro's `\Titel` en `\Spreker`. De macro `\Titel` expandeert tot `\titel`, mits dat veld bestaat. Als dat veld niet bestaat, expandeert `\Titel` tot **titel volgt**. De titel is in ons geval bekend, dus `\Titel` expandeert tot **Vijftig jaar VVS: nog eenmaal de magere jongelieden**. De veldnamen kunnen zo eenvoudig verstekwaarden krijgen. Evenzo expandeert de macro `\Spreker` tot `\spreker`. Echter, de macro `\spreker` expandeert niet tot tekst, maar tot `*zwet* *`. Dit is een verwijzing die onder meer de expansie van `\persoon` behelst. Interessant aan de expansie van `\persoon` is dat `\Titels` expandeert tot `\profdr`, dat gedefinieerd is als `prof. dr. .` Op deze wijze is eenduidigheid in titulatuur ook eenvoudig te bewerkstelligen. Dit veld kunnen we eveneens gebruiken bij de adressering van een brief:

```

\newcommand{\Aanhef}{%
\IfField\Titels\Exists
\IfExistingField\LooksLike
\titels\profdr;%
\renewcommand{\Aanhef}{\bf%
Aan de hooggeleerde heer
\titels\Vlts\naam}}%
\Fi
\Fi

```

Hier vergelijkt δTEX de inhoud van het veld `\titels` met `\profdr` zonder `\profdr` te expanderen. In ons voorbeeld zal `\Aanhef` uiteindelijk expanderen tot **Aan de hooggeleerde heer prof. dr. W.R. van Zwet**. Uiteraard kan een vergelijkbare macro worden gebruikt om te controleren of een brief wordt gericht aan een vrouw, een man, of een man of een vrouw.

De verwijzing `*hfd1* *` levert dus uiteindelijk **‘Vijftig jaar VVS: nog eenmaal de magere jongelieden’, prof. dr. W.R. van Zwet**. In de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -code van het programmaboekje wordt alleen `*hfd1* *` gebruikt, en niet de ‘harde’ tekst waarin `*hfd1* *` tenslotte resulteert. Dit heeft als groot voordeel dat wijzigingen in de gegevens slechts één keer ingevoerd hoeven te worden om in het uiteindelijke zetwerk alle keren te worden doorgevoerd.

Naast het programma bevat het programmaboekje ook de samenvattingen van de voordrachten. Bij de samenvattingen wordt niet alleen een korte beschrijving van de voordracht afgedrukt, maar ook uitgebreide adresgegevens van de spreker. De expansie van `\lezing` moet zó worden gewijzigd dat het bestand met de tekst van de samenvatting wordt ingelezen, en de expansie van `\persoon` gaat nu ook de adresgegevens afdrukken.

```

\renewcommand{\persoon}{%
\renewcommand{\Titels}{}%
\IfField\Titels\Exists
\renewcommand{\Titels}{\titels}%
\Fi
...
\renewcommand{\Email}{}%
\IfField\email\Exists
\renewcommand{\Email}{\texttt{\email}}%
\Fi
\Titels\Vlts\naam, \Vakgroep
\Straat\Pcode\Plaats\Email
}%
\renewcommand{\lezing}{%
\renewcommand{\Titel}{titel volgt}%
\IfField\Titel\Exists
\renewcommand{\Titel}{\titel}%
\Fi
\renewcommand{\Spreker}{spreker onbekend}%
\IfField\spreker\Exists
\renewcommand{\Spreker}{\spreker}%
\Fi
\IfField\tweedeauteur\Exists
\renewcommand{\Spreker}{\spreker,
\tweedeauteur}%
\Fi
\renewcommand{\Samenvatting}{Er is helaas
geen samenvatting beschikbaar}%
\IfField\samenvatting\Exists
\renewcommand{\Samenvatting}{%
\input abstract/\samenvatting}%
\Fi
{\bf '\Titel', \Spreker}\smallskip
\Samenvatting\bigskip
}

```

De uiteindelijke lijst met samenvattingen genereren we met het commando `\Filter lezing\Type\lezing`. Alle `\lezings` in `lezing.dat` expanderen nu volgens de bovenstaande definitie. Als alle gegevens in `lezing.dat` zijn bijgewerkt, zal van geen enkele lezing de samenvatting worden vergeten. Indien van een lezing geen samenvatting beschikbaar is, wordt een tekst van die strekking afgedrukt. Als het veld `\samenvatting` wel bestaat, zoekt δTEX het aangegeven bestand in de subdirectory `abstract` en leest het bestand in. De tekst in dat bestand moet natuurlijk wel correcte $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -code zijn. Merk op dat `\persoon` nu ook de adresgegevens van de spreker (of een spreker en zijn of haar co-auteurs) afdrukt. Voor elke expansie van `\persoon` wordt nagegaan of alle relevante velden beschikbaar zijn. Er wordt niets (of een verstektekst) afgedrukt als dat niet het geval is. In ons voorbeeld is het resultaat van `hfd1` nu

**‘Vijftig jaar VVS: nog eenmaal de magere jongelieden’,
prof. dr. W.R. van Zwet, Vakgroep Mathematische
Statistiek, Niels Bohrweg 1, 2333 CA Leiden.**

<zetsel abstract/vzwet.tex>

Uit het bovenstaande kan de indruk ontstaan dat het samenstellen van een programmaboekje niet veel eenvoudiger wordt door het gebruik van $\delta\alpha\text{T}_{\text{E}}\text{X}$. In plaats van het zetten van teksten zou men nu veel tijd kwijt zijn met het schrijven van macro's. Dit is een misvatting: een beperkt aantal macro's genereert een compleet programmaboekje. Juist door het scheiden van gegevens en

tekst wordt grote consistentie in het eindresultaat bereikt en is het eenvoudig om lezingen die één voor één worden aangemeld, bij de organisatie te verwerken. Het uiteindelijke zetwerk wordt door $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$ en $\delta\alpha\text{T}_{\text{E}}\text{X}$ gedaan; met een zeer beperkt aantal macro's en regels tekst is in een handdraai een programmaboekje gegenereerd (de $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$ -file voor het programma van het jubileumcongres van de VVS bestond uit zo'n 300 regels, de uiteindelijke omvang van het programmaboekje was 44 pagina's). Het uiteindelijke werk is niet het typen van het boekje, maar het vullen van de bestanden `lezing.dat` en `spreker.dat`.

Adobe Supra: Adobe's High-Performance Raster Image Processor (RIP) Architecture for Print-on-Demand Systems

Wiegert Tierie

Adobe Systems, Amsterdam

During the last decade, the graphic arts and publishing industries have witnessed profound changes nearly tantamount to the revolution spawned by Johannes Gutenberg with his printing press in 1455. Dynamic technological advances in personal computers, workstations, open systems, desktop software applications, and networks produce finished documents faster and at lower prices. Document authoring and distribution tools from Adobe and other vendors enable organizations to create, access, distribute, and print pages and documents electronically. Indeed, the technologies are converging to create fresh opportunities and new markets, thus unleashing the Age of the Digital Document.

A significant number of pages are now produced completely digitally. The arrival of digital printing, which eliminates film altogether, dramatically affects the prepress and production world. With paper and distribution costs rising, the advent of the digital press makes quick turnaround of short-run, high-quality color jobs a real possibility and supports customized printing in which each succeeding page on the press can be different from its predecessor. Distributed printing — in which materials are sent electronically over telephone, the Internet, or satellite links for printing nearby destinations — saves distribution costs.

Developments such as digital presses are revolutionary. Their adoption rate by the printing industry may be a function of the rate at which printing companies are willing and able to make capital investments and re-engineer workflow. Companies looking for a more evolutionary approach can turn to the new generation of large-format imagesetters and digital platemakers. These devices can help reduce labor and materials costs, shorten production cycles, and reduce the environmental impact of chemical processes. All these factors can assist printing companies in remaining more competitive with digital presses for fixed-content jobs.

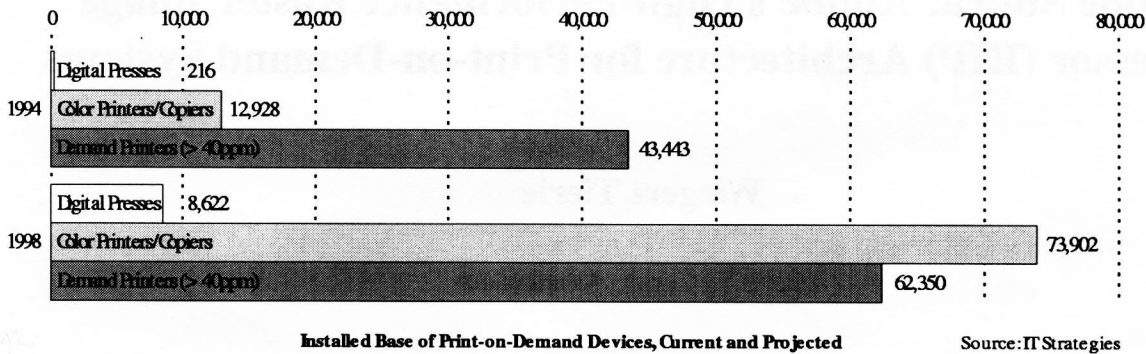
Adobe and its PostScript page description language OEM customers recognize the potential impact that large-format imagesetters, digital platemakers, and presses will have on the printing and publishing industries, and they are working to make those benefits widely available.

Market Outlook

As with any new market, the business models, cost structure, and real benefits of digital presses continue to evolve. The business opportunities for digital presses are ripe and potentially huge, but they are largely untapped at this point. That will change.

According to the market-research firm BIS Strategic Decisions, the market for electronically produced, short-run color pages is expected to grow significantly. In 1993, the group noted that the short-run commercial printing market for both electronic monochrome and color printing of 5,000 pages or less was \$31 billion. By 1998, BIS projects that 49 percent of the short-run commercial printing market will be full-color output, 29 percent will be two color, and 22 percent will be one color.

Two years ago, digital presses were a concept. But by the end of 1994, more than 200 had been installed worldwide. IT Strategies, a market-research firm, projects that this number will grow to more than 8,600 devices worldwide by the end of 1998. But these are not the only contenders in the print-on-demand market. IT Strategies predicts that the number of color printer/copiers will increase from 13,000 to about 74,000 in the same time period, a growth rate of 55 percent per year. Also, high-speed printers (more than 40 pages per minute) will increase from 43,000 to 62,000. Many of these devices will be upgraded to support additional colors and graphic arts capabilities (they will contain the PostScript language).



It is difficult to predict the impact of the Internet. World Wide Web data delivery increases by a factor of 10 every 9 or 10 months. This trend may accelerate as mechanisms are put in place to make commercial publishing viable on the Web.

In short, there is a convergence of three market thrusts: short-run color, demand printing, and distributed printing. Short-run color has been on commercial print buyers' wish lists for years. Demand printing can be thought of as the high-volume printing coming from MIS departments and data centers. And distributed printing is the new 'distribute and then print' metaphor enabled by our global communications network.

Below is a sample of the implications to these initiatives.

- From the BIS statistics above, the color content of pages will continue to increase; Adobe believes, by inference, that the graphical content will also.
- Documents will be formatted for multiple delivery options, such as print, on-line, and CD, and this activity will continue to fuel the growth of these alternative delivery mechanisms.
- There will be an explosion in the number of publishers on the Internet. Anyone with a PC, a modem, and a page layout program can become a publisher.
- There will be increased deployment of large-scale, graphical document databases. Whether centralized or distributed, document management and search techniques will be the keys to making the information usable and accessible.
- Demand for color printers will be driven by the increase in color pages and the broad use of color on the World Wide Web, an increasingly important source and repository for digital documents.
- System-level, device-independent color through International Color Consortium (ICC) standards will become the preferred way to obtain reliable

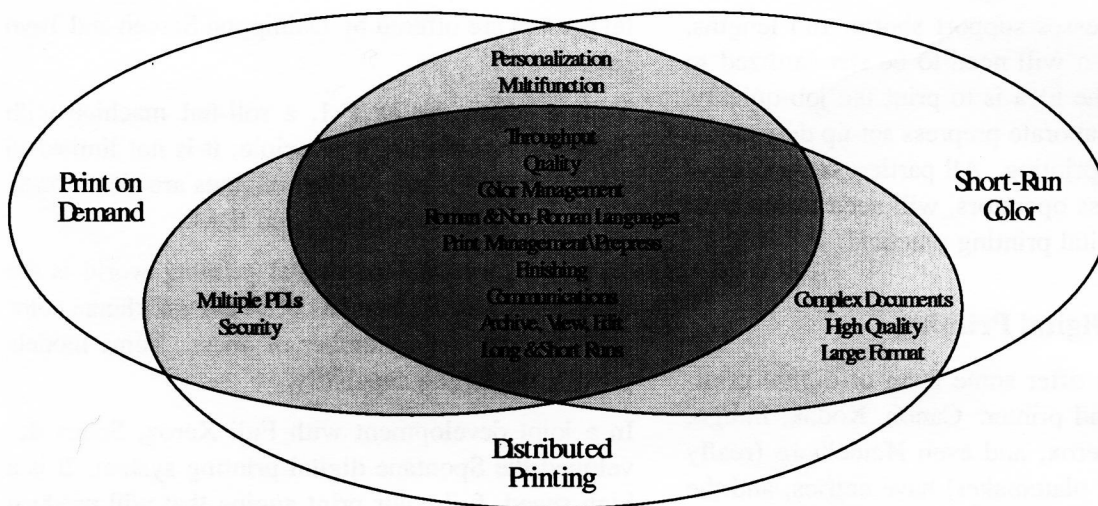
and predictable color, especially when it is not known which printing device or technology will be used at the destination. (Adobe was one of the founders of ICC. The ICC standards are compatible with PostScript Level 2, which was defined three years before the standards themselves.)

- To reduce transmission time on networks, data and image compression will be used.

As digital documents become the norm, the demands of printing markets that were previously separate now overlap and intersect. The print-on-demand market, short-run color market, and distributed printing market now have similar requirements for factors such as throughput, quality, color management, finishing, and print management. Indeed, these requirements represent the dimensions to the challenge. The need is clear for an integrated system that will enable users to manage document printing and distribution efficiently in the new print-on-demand world.

Who Will Benefit From Digital Print Advances?

To fully realize the benefits of digital print advances, one must embrace a totally desktop-based digital workflow. Digital documents alone provide numerous advantages. Typically, desktop systems require less space and power than traditional graphic arts processing equipment, and they are more cost effective than proprietary color electronic reproduction systems. Traditional color reproduction requires tremendous quantities of expensive film and chemical processing. The closer one gets the image to the actual press, the better the image quality and the greater the reduction in labor content, supply costs, and environmental impact. All of these factors lead to reduced cost. But the digital document also implies reusability, which will further reduce costs.



The Merging and Intersection of Key Markets

One of the hidden benefits of the digital document is what it can imply in the review and sign-off phase of projects. Many companies are already using Adobe Acrobat and the Adobe Portable Document Format (PDF) for this purpose. PDF, with roots in the PostScript language, supports the full range of high-quality typography, graphics, images, and color. PDF files can be created from virtually any desktop application. Adobe provides Acrobat readers on all major desktop computers. PDF files are ideal for passing from one location to another because they are concise (image and data compression) and because Acrobat contains sophisticated font substitution technology that produces accurate renditions of fonts even when the destination computer does not have the document's original fonts. Reviewers can see a graphically accurate version of the job. The ability to make annotations and to respond over computer networks and modems significantly reduces the time for sign-off cycles.

But anyone who uses paper will benefit from digital printing technology. Further, anyone needing short-run production of 5,000 or fewer pages or copies, quick turnaround, and some degree of customization will want to turn to a digital press for output. Digital presses and new-generation presses with on-board, direct-to-plate capabilities now make it practical to print short runs, which were not economically feasible before because of the make-ready for an offset press. The trade-offs are between getting exactly what you want and the number you want versus the speed and low unit cost of volume printing, but with the associated schedules, inventory, obsolescence, and waste issues.

For businesses and consumers, digital technology offers the opportunity to print customized short runs of color documents, potentially on any print medium. The Indigo Omnius system, for instance, can print on packaging materials such as soda cans, foam, mylar, and aluminum. Real estate professionals, marketers, advertising agents, and even restaurant owners can personalize their materials in a cost-effective manner. Imagine the impact if your realtor showed you a flyer of your 'dream house' with your family standing in the front yard. The possibilities are endless for targeted marketing, direct mail, and publishing.

Digital presses will also encourage 'distribute and then print' applications. For example, book publishers may no longer need to print many thousands of books and then absorb expensive inventory and distribution costs. Files can be distributed digitally and printed in the quantity needed for each local market. This method becomes particularly important for printed materials distributed internationally so that shipping and customs delays can be avoided.

Traditional commercial printers can expect to see benefits when this technology is applied to large-format imagesetters and platemakers. Not only does the added processing power allow shorter production cycles and 'rush' processing, but it can help address a potentially costly problem. When a printing plate is defective (often not discovered until the job has started running on the press), a new plate must be generated rapidly while the press is idle. Every minute that the press is idle represents lost profits to the printer. Often a plate is rendered defective because a telephone number or a price is wrong. In these cases, the appropriate flat must be re-RIPped.

Because digital presses support shorter run lengths, prepress preparation will need to be standardized to be economical. The idea is to print the job quickly, so time spent on elaborate prepress set-up defeats the purpose of digital printing. All parties, from designers through prepress operators, will need to work together to make digital printing practical.

Contenders in Digital Printing

Several companies offer some form of digital printing press or demand printer. Canon, Kodak, Indigo, Scitex, Xeikon, Xerox, and even Heidelberg (really an on-press digital platemaker) have entries, and the PostScript language is the common denominator for page description. Indigo has continuously set the standard for print capabilities, particularly with color, paper stocks and weights, media types, and finishing options. Xeikon was the first digital press system for roll-fed applications. Adobe is the largest supplier of RIP software with its Configurable PostScript Interpreter (CPSI) for the broad set of technologies that service print-on-demand applications.

Additional technologies offer useful subsets of the total print-on-demand feature set and are moving to increase their capabilities. Color printer/copier vendors are increasing speed and quality. Some already have duplexing, or printing on both sides of the paper. These vendors include Canon, Kodak, and Xerox. Often the RIP technology for this class of device has been provided by COLORBUS, Electronics for Imaging (EFI), and Radius using Adobe CPSI. Demand printer manufacturers already have fast machines and are trying to add color, increase resolution, and provide graphically rich page formatting through the PostScript language and the Adobe Portable Document Format (PDF). These vendors include IBM, Kodak, Nipson, Siemens, and Xerox. Over the past nine months, new color laser printer product introductions have come from Apple, Hewlett-Packard, Tektronix, and Xerox.

Indigo makes three machines that can be categorized as digital presses. The E-Print 1000 uses liquid inks and is sheet fed. It uses a single cylinder for up to six revolutions per side in which a different color is applied on each revolution. As many as 100 A3-size pages can be printed, folded, and stapled into booklets. The Mobius is a duplexed roll-fed, electro-ink device. The Indigo Omnius, mentioned earlier, will print on packaging materials. In Japan, Indigo digi-

tal presses are offered by Dainippon Screen and Toyo Ink.

Xeikon offers the DCP-1, a roll-fed machine with eight toner stations. In principle, it is not limited to a fixed paper length. Xeikon engines are also offered by Agfa, AM International, and IBM.

Canon's entry into the digital printing world is the CLC. This product line has been the workhorse color printer/copier for a number of years. Some models provide duplexing capability.

In a joint development with Fuji Xerox, Scitex developed the Spontane digital printing system. It is a high-speed, full-color print engine that will produce up to 40 full-color A4 pages per minute. Scitex also offers Scitex Digital Printing, a low-resolution color digital press that can run at 200 feet per minute using continuous inkjet technology.

Heidelberg has entered the market with two on-press platemaker systems: the GTO-DI and Quickmaster DI. Although good for short-run jobs, they are not true digital presses, because they still use a master plate to print all the copies. The Quickmaster actually has plate material in the machine so that when one plate is done, it rolls forward and then, for the next job, the next flat is imaged right on the machine.

Supra-Adobe's High-Performance System Architecture

Adobe Systems has developed a strategy for meeting the needs of this emerging on-demand market — a strategy that includes a new system architecture called 'Supra', an extension of the PostScript language and PDF standards, and collaboration with key printer OEMs and workstation system vendors. Supra takes a systems perspective within an integrated workflow, rather than addressing requirements piecemeal and ending up with a series of point solutions that may not mesh with customer systems environments.

The architecture is supported by all the desktop prepress tools needed to prepare digital documents for production. Adobe and others provide desktop and server solutions for preflight and color separation, trapping, imposition, and high-resolution picture replacement. Adobe believes that prepress functions should be available for 'desktop server and RIP' integration to support the broad spectrum of workflows while providing consistent results.

This architecture is applicable to not only digital presses, demand printers, and color printer copiers, but also to large-format imagesetters, proofers, and platemakers. All these devices share a common requirement: to deliver a large number of color separations per unit of time to the marking engine in the right order. Currently, digital presses tend to produce lower resolution but higher page speed, while imagesetters operate at higher resolution but lower page speed.

The architectural objectives include:

- **Scalability**
Adding processing power and system resources to increase throughput.
- **Configurability**
Selecting and including only those pieces needed to meet the system requirements. (In some cases, this implies including some of Adobe's Supra modules within existing systems.)
- **Multiplatform**
Implementation of the architecture using different computer platforms.
- **Openness**
As interfaces and data structures are solidified and verified for system performance, Adobe will open them up for OEM and third-party development.
- **Quick to market**
Efforts will be made to use existing Adobe and OEM technology to bring products to market sooner.
- **Extensibility**
Evolving the architecture for greater performance and functionality.

End user expectations guide the Adobe architectural approach. First and foremost, the architecture must be *high performance*; it must support *fast turnaround* and *production volumes* whether it produces many copies of static content or many unique pages. The maximum run length should be limited by the cost of this print technology versus the cost of printing using a different technology, such as offset lithography. Each page should be customizable with everything from different textual content to personalized images and charts. Additionally, the architecture must enforce guaranteed delivery, implying the need for sophisticated error handling and recovery.

The above diagram shows a portion of the Adobe Supra architecture. To achieve the desired processing rates, Supra employs a multiprocessor architecture. For this scheme to be successful, independent pages must be passed to each RIP process.

Key features of the architecture include:

- PostScript and PDF workflows
- Enforced PostScript page independence so that multiple processors can work on the job simultaneously,
- Overlapped RIPping and output
- High-speed merging of graphically rich customized data
- Interfaces to non-PostScript printing environments, page description languages, and databases
- Job control, sophisticated error handling, and control of finishing options
- Automation of prepress processing and 'reflow' document assembly through workflow systems
- Viewing, editing, and archiving with PDF files

The Role of Adobe

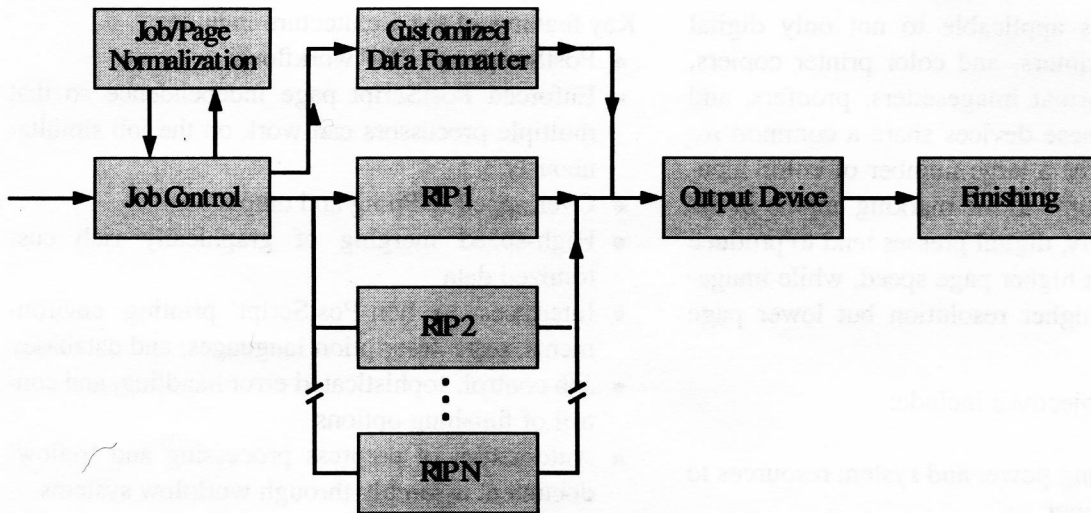
The PostScript language will continue to be the primary format for print production. Because it is a true programming language, pages can be arbitrarily complex and sophisticated. The Adobe Portable Document Format (PDF) is increasingly becoming a standard for cross-platform document communication, viewing, and archiving. Adobe is committed to adding significant functionality to the PostScript language and PDF standards to serve market needs.

In collaboration with a number of its OEM suppliers of output devices and workstation system vendors, Adobe is evolving the Supra architecture and is establishing interfaces and data standards to develop the print-on-demand market.

Adobe is working on an open standard for variable data for all of its applications, drivers, and RIP technology, and will offer its products broadly. The company plans to integrate standards that specify press functions such as color control as well as finishing options.

The Future

Digital presses will surely reshape the printing and publishing industries, just as electronic platforms have revolutionized traditional page layout and prepress. The potential impact should not be ignored. The tools and technologies for viable products exist, but there is still some distance between the current array of products and standardized, automated production. Digital printing, to be effective, should not be considered a stand-alone solution. A total sys-



tems approach is needed, with workflow and printing strategies considered.

Just as RIPs are an essential element in the equipment architecture today, finishing capabilities will expand future options. Control of the press and post-press functions will be specified as a prepress step prior to RIPping. Eventually, all aspects of production will be controlled from the desktop.

But the net benefit of digital printing will be a printing industry that is more responsive to the needs of its customers.

The digital document will become synonymous with getting the content, image, customization, and quantity when and where you want it.

Computergestuurd Zetten of Electronisch publiceren? Nieuwe trends in wetenschappelijk publiceren*

Philip Taylor

Abstract

W ciągu ostatnich 15 lat skład tekstu wspomagany komputerowo całkowicie wyeliminował stosowanie tradycyjnych technik drukarskich. Obecnie stoimy przed jeszcze bardziej radykalną rewolucją technologiczną, jaką stanowi pojawienie się wydawnictw electronicznych (electronic publishing - w skrócie e.p.).

W przeciwieństwie do składu komputerowego i tradycyjnego, publicacje electroniczne w ogóle nie wykorzystują papieru, nośnikiem informacji staje się ekran komputera.

Potencjalne korzyści płynące z zastosowania tej nowej metody publikacji wydają się oczywiste: znaczna redukcja kosztów i niemal natychmiastowa dystrybucja do zainteresowanych odbiorców. Nie można jednak pominąć i wad, takich jak: możliwość bezprawnego kopiowania, łatwość plagiatów i nielegalnej redystrybucji.

Obie wspomniane metody publikacji różnią się w sposób zasadniczy: o ile systemy składu tekstu z góry narzucają postać końcową strony, to systemy publikacji electronicznych stwarzają jej bardziej niezależną reprezentację, w której ostateczna forma zależy raczej od programu prezentującego dokument (tzw. viewera, czy po polsku - przeglądarki).

W niniejszej pracy omówione są najkowsze osiągnięcia zarówno w dziedzinie składu komputerowego jak i publikacji electronicznych. Wskazuje się również na różnice tych dwóch metod rozpowszechniania.

Een korte geschiedenis van T_EX

Tot ongeveer 15 jaar geleden werd het zetwerk vrijwel volledig genegeerd door de overgrote meerderheid van alle wiskundigen, wetenschappers en geleerden in het algemeen: manuscripten werden voorbereid met een typemachine, de meer verzochte tekens (voor wiskundigen betekende dat vrijwel alle symbolen) werden later met de hand ingetekend, en het geheel werd gewoon verstuurd naar de uitgever. Een tijdje later kwamen dan de voorproeven terug, verbeteringen werden genoteerd in de kantlijn, en vervolgens werd het geheel weer teruggestuurd naar de uitgever. Eenzelfde maar kortere cyclus volgde vervolgens voor de huisproeven, en uiteindelijk verscheen het in de door de auteur gewenste vorm in het voltooide boek. Op geen enkel punt was er een direct contact tussen de auteur en de zetter, en feitelijk was de eerste persoon zich ook vrijwel onbewust van het bestaan van de tweede.

De zetter was zich echter maar al te goed bewust van het bestaan van de auteur: wiskundig zetwerk werd traditioneel in de zeterswereld altijd aangeduid als 'strafwerk', omdat het berucht moeilijk is om het correct uit te voeren. In de tijd waarin een collega makkelijk in staat was tot het zetten van tien pagina's tekst, was de wiskundige zetter net in staat tot het zetten van een enkele pagina, en zelfs dan wist hij al dat het heel waarschijnlijk was dat het geheel meer dan één keer overnieuw gezet zou moeten worden, aangezien wiskundigen erg gretig zijn in het uitvinden van nieuwe symbolen als de bestaande net niet helemaal lijken

te voldoen. Aangezien de zetter dat symbool nog nooit eerder gezien zou hebben, zou hij (heel redelijk) aannemen dat het gewoon een slecht ingetekende versie van een ander symbool betrof, één die hij wel herkende, en vervolgens dat symbool gebruikt hebben ...

Het lag voor de hand dat enkele van de meer opletende auteurs begonnen te experimenteren met computertechnologie zodra die meer algemeen beschikbaar werd, en voor een tijdje was de academische wereld er schijnbaar van overtuigd dat alle problemen opgelost konden worden als het mogelijk zou zijn om nog een paar extra tekens toe te voegen aan het margrietwiel van de Diablo printer. Er waren zelfs bedrijven die gespecialiseerd waren in het aanpassen van margrietwielen door het vervangen van een kennelijk onnodig teken door één waarvan de eigenaar dacht dat het volstrekt onmisbaar was. Natuurlijk was deze aanpak gedoemd te mislukken: Het is net zo onmogelijk om wiskundig zetwerk te doen met een vaste set van 144 tekens als dat het is met een set van 128 tekens, en ondanks de beste pogingen van de betrokkenen verdween de margrietwiel-printer toch al snel naar de prullenbak.

Gelijktijdig met deze pogingen, begonnen de producenten van matrix-printers een zekere invloed te krijgen. Met een 7×5 matrix, is er een potentieel aantal verschillende tekens beschikbaar van

$$\sum_{i=0}^{35} \binom{35}{i} = 2^{35}$$

*Dit artikel werd oorspronkelijk gepresenteerd als een lezing voor een gecombineerde bijeenkomst van de Poolse Wiskundige Vereniging en het EmNet Projekt (Euromath Trust) in Torun, Polen, 1995. Sindsdien is het gepubliceerd in het engels in GUST (Grupa Użytkowników Systemu T_EX); Biuletyn Zeszyt 6, pp. 12-27, een poolse vertaling verscheen in *Pro Dialog*, Augustus 1996, en de engelse versie werd wederom gedrukt in TUGboat 17 (1996), pp. 367-381.

Het artikel is gherdrukt met toestemming van de auteur, editors en de organisatie van de oorspronkelijke bijeenkomst. Deze vertaling in het nederlands is gemaakt door Taco Hoekwater, eventuele manco's in het taalgebruik zijn zijn verantwoordelijkheid.

(inderdaad een enorm aantal). Helaas zijn er daarvan een aantal vrijwel onmogelijk van elkaar te onderscheiden: een enkele stip op de coördinaten (4,3) lijkt verbazend veel op een andere stip op de coördinaten (4,4), zelfs voor de meest oplettende lezer (Ik geloof dat er een totaal van 33 034 338 305 *verschillende* tekens mogelijk is, in tegenstelling tot een totaal van 34 359 738 368 totale tekens. Daarbij wordt een tekens als *verschillend* aangeduid indien het niet eenvoudigweg wordt verkregen door het horizontaal, verticaal of in beide richtingen verschuiven van een ander teken. Deze getallen zijn gebaseerd op een analyse uitgevoerd door Dr. Warren Dicks van de Autonome Universiteit van Barcelona). Daarbij komt dat de kwaliteit van de uitvoer van een 7×5 matrix printer zo ongelooflijk slecht is dat *geen enkele* poging gedaan zou moeten worden om er boeken mee te maken – een goed advies dat toendertijd helaas zelden werd gevolgd.

Uiteraard moest er geschikte software worden geschreven om die nieuwe technologische uitvindingen te kunnen gebruiken, en met name de Unix wereld besloot te standaardiseren rond het programma ROFF en zijn afgeleiden: NROFF, TROFF en uiteindelijk DITROFF kwamen allemaal aan bod. Helaas bood geen één van de uit ROFF afgeleide programma's de mogelijkheid om direct wiskunde te zetten, zodat aparte programma's gebruikt moesten worden zoals EQN en TBL om de wiskundige functionaliteit toe te voegen. Er waren ook wel commerciële systemen, die bijvoorbeeld gebruikt werden voor het zetten van publicaties zoals de *Transactions of the American Mathematical Society*, maar die waren zowel duur als ondoorgrondelijk, met gebruikmaking van een non-mnemonische syntax voor het beschrijven van de mogelijke wiskunde constructies.

Gelukkig (zoals terugkijkend overduidelijk zichtbaar is), was er op zijn minst één uitmuntende wiskundige die geloofde dat iets beters niet alleen gemaakt kon, maar ook gemaakt *moest* worden; en omdat hij niet alleen een wiskundige was maar ook een computer-wetenschapper, besloot hij het zelf te schrijven. Zijn naam was Knuth, en zijn programma was T_EX.

En toch, we danken het alleen aan een gelukkig toeval dat T_EX geboren werd. In die tijd werkte Knuth aan zijn *opus magnum*, een serie van 7 boeken getiteld *The Art of Computer Programming*, en in 1977 bleek dat de populariteit van de eerste delen zo groot was dat deel 2 al aan een herdruk toe was. De timing hiervan viel helaas zó dat hoewel de eerste editie nog gedrukt was met de traditionele loodmethode, de tweede editie gedrukt werd met een van de eerste fotozetters [aan de lezer: door het hele artikel gebruik ik het woord *zetter* voor zowel de *persoon* die het zetwerk uitvoert, als voor de *apparatuur* die daarvoor gebruikt wordt. Ik hoop dat het altijd duidelijk is welke van de twee betekenissen bedoeld wordt, aangezien er geen ander woord is dat eenvoudig en eenduidig gebruikt zou kunnen worden om één van de twee betekenissen te vervangen]. En hoewel de nieuwe fotozetter *in theorie* in staat was tot het leveren van gelijkwaardige, of zelfs betere, uitvoer dan de traditionele loodmachine die eerder ge-

bruikt werd, liet het resultaat veel te wensen over. Knuth, als wiskundige en wetenschapper, was ervan overtuigd dat de fout niet in het apparaat zelf lag, maar in de software die gebruikt werd om het te besturen. En liever dan zijn levenswerk te zien verschijnen in een tweede-keus formaat, besloot hij een klein gedeelte van zijn professionele leven te besteden aan het schrijven van een verzameling programma's die *wel* in staat zouden zijn het volledige potentieel van de fotozetter te gebruiken. Toen hij deze moedige beslissing nam kon hij nog niet vermoeden dat het uiteindelijk minstens tien jaar zou gaan duren, in plaats van het ene jaar dat hij ervoor geraamd had, dit ondanks het feit dat hij binnen dat ene jaar toch beslist een demostreerbare werkende versie had.

De eerste gepubliceerde referentie naar T_EX is waarschijnlijk *Mathetical Typography*, uitgegeven als rapport STAN-CS-78-648 door de computer wetenschappelijke afdeling van de Stanford Universiteit; in de bibliografie daarvan geeft Knuth de uiteindelijke referentie op als: *Tau Epsilon Chi, a system for technical text*. Op dat moment was die nog 'in bewerking' en nu helaas uit de handel. Voor lezers die geïnteresseerd zijn in het onderwerp, is het genoemde artikel beslist het lezen waard, en de bibliografie alleen al maakt het een waardevolle aanwinst. Het werd heruitgegeven in het *Bulletin of the American Mathematical Society*, en in die vorm zou het nog steeds beschikbaar moeten zijn.

T_EX was zowel typisch voor als afwijkend van de programma's uit die tijd: het was typisch aangezien het volledig script-georiënteerd was, een gevolg van het feit dat het programma ontstond voordat er een wijd-gebruikte grafische werkomgeving beschikbaar was; het was afwijkend in die zin dat het een volledige programmeerbare *macro* programmeertaal bevatte, waarin er geen gereserveerde woorden waren, en waarin zelfs eenvoudige tekens in staat waren dynamisch van betekenis te veranderen. Zodoende bestond een T_EX document uit zowel de tekst die gezet moest worden als uit de commando's die daarvoor zorgden, en alleen T_EX zelf was eenduidig in staat om te beslissen of iets beschouwd moest worden als 'programma' of als 'tekst'.

Ondanks het feit dat het feitelijk werd geschreven voor het bereiken van één enkel doel – het zetten van Deel 2 van *The Art of Computer Programming* – ontwikkelde T_EX al snel een eigen leven, en al vlug werd het de *de facto* standaard voor zetten binnen een groot gedeelte van Stanford University. Al snel was T_EX ook daarbuiten beroemd, en in 1980 onstond de T_EX Users Group, met bestuursleden van ver buiten de grenzen van de faculteit van Stanford. De American Mathematical Society was gerepresenteerd binnen dat bestuur, en de verbanden tussen Knuth en het AMS waren hecht: Knuth droeg het T_EX logo over aan het AMS die daarna een trademark aanvroegen als bescherming tegen niet-officiële wijzigingen aan het programma – helaas werd het verzoek afgewezen wegens een eerdere toewijzing van TEX aan Honeywell, maar ondanks het gebrek aan een officiële registratie zorgen de hoge standing en hoge waardering voor Knuth ervoor dat

het \TeX logo (en zijn niet-gezette vervanger, TeX) alomerkend en gerespecteerd wordt.

Binnen een paar jaar werd duidelijk dat de originele implementatie van \TeX nog een paar dingen te wensen overliet, zowel in termen van functionaliteit als in termen van portabiliteit, en Knuth besloot beide bij te werken door \TeX compleet te herschrijven. Deze keer besloot hij SAIL ('Stanford Artificial Intelligence Language') te vermijden, en in plaats daarvan gebruik te maken van de veel algemener gebruikte programmeertaal Pascal. Om de portabiliteit nog groter te maken, adopteerde hij alleen een precieze subset van Pascal, met gebruikmaking van alleen die mogelijkheden waarvan hij erop vertrouwde dat ze in elke Pascal implementatie aanwezig (of eenvoudig na te maken) zouden zijn. Tevens besloot hij daarbij meteen gebruik te maken van deze kans om het programma te schrijven in een vorm die hij 'beletterd' noemde: hij wilde dat mensen in staat zouden zijn om de broncode van \TeX te kunnen lezen op dezelfde manier waarop ze een boek zouden lezen, zodat ze daarbij in staat zouden zijn om te profiteren van een groot produkt van software-ontwikkeling, gepresenteerd in een leesbare vorm. Wederom besloot Knuth dat hiervoor geen geschikte hulpmiddelen beschikbaar waren, en wederom boog hij daarom af van het hoofdproject voor het ontwikkelen en implementeren van het WEB concept, met de twee daarbij behorende programma's TANGLE en WEAVE.

Een WEB programma bestaat uit een hoogst gestyleerd dialect van Pascal, met grote tussengevoegde commentaren die de bedoeling en functie van de elementen en modules van het programma uitleggen (Ik vermoed dat Knuth dit zou ontkennen, en in plaats daarvan zou zeggen dat een WEB programma bestaat uit een hoogst uitgebreide beschrijving van de werking van het programma, hier en daar onderbroken met stukjes Pascal die die functionaliteit implementeren; en ik vermoed dat hij vrijwel zeker gelijk zou hebben!). Door het toe te staan de stukken van het Pascal programma in een willekeurige opeenvolging te presenteren (in tegenstelling tot de strenge regels over de volgorde die worden gesteld in de Pascal standaard), geeft WEB de programmeur de mogelijkheid om de onderdelen van een programma te presenteren in een natuurlijke en logische volgorde, in plaats van de kunstmatige opeenvolging die wordt afdwongen door de Pascal ontwerp-eis van 'efficiënte compileerbaarheid'. Het is de taak van het programma TANGLE om ervoor te zorgen dat de brokstukken worden samengevoegd op de manier die geëist wordt door Pascal, en het is de taak van WEAVE om ervoor te zorgen dat de commentaren samen met de brokjes code omgewerkt worden in een vorm die geschikt is om direct door \TeX gezet te worden.

Zodoende werd \TeX voor de eerst keer zelf-verwijzend: om de Pascal code te krijgen uit de WEB broncode, was een werkende versie van TANGLE nodig, en om een leesbare versie van de code te krijgen was een werkende versie van WEAVE nodig, maar beide programma's zijn zelf geschreven in WEB, dus om een werkende versie van TANGLE te maken is een werkende versie van TANGLE

nodig, *ad infinitum*. Natuurlijk is 'bootstrapping' (zoals de technische term luidt) al lang bekend en begrepen binnen de computerwereld, en de verwachting was dat 'met de hand compileren' van TANGLE uit de WEB broncode daarvan ruim binnen de grenzen van de mogelijkheden van de 'gemiddelde implementator' zou liggen. Ik herinner me echter maar al te goed het trauma dat een collega meemaakte toen hij probeerde dat 'bootstrappen' zelf te doen...

Tijdens de her-implementatie herschreef Knuth vrijwel het gehele \TeX programma: hij had veel geleerd over de grenzen van \TeX tijdens die eerste jaren van gebruik, en in 1982 was er een volledige herschreven \TeX ontstaan. Deze versie van \TeX (vaak aangeduid als \TeX 82 om het verschil duidelijk te maken met de eerdere versie die analoog hieraan werd aangegeven als \TeX 78) werd snel overgedragen naar een wijde verzameling machines, en is misschien wel het wijdst verspreide programma in de wereld op dit moment, beschikbaar als het is voor elk type systeem vanaf de kleinste PC tot de grootste super-computer. De vrijwel algemene acceptatie van \TeX als *het* standaardpakket voor computer-zetten is vrijwel zeker het resultaat van een grote verzameling van positieve eigenschappen: De broncode van het programma, en van de meeste implementaties, is gratis beschikbaar danwel voor een minimaal bedrag dat uitsluitend dient ter compensatie van de kosten van het materiaal waarop verspreiding plaatsvindt; het programma is vrijwel bug-vrij, een claim die tot-voorkort ondersteund werd door Knuth zelf met een cheque die uitgereikt werd voor iedere bug die gemeld werd, waarbij de waarde van de cheque elk jaar verdubbelde sinds het systeem werd ingevoerd (hij biedt nog steeds een cheque aan, maar de waarde verdubbelt niet meer elk jaar, omdat hij heeft geschat dat anders over niet al te lang het een groter bedrag zal zijn dan de federale reserves...); het programma is erg stabiel (er waren vrijwel geen belangrijke veranderingen gedurende de periode 1982-1990, en evenzo zijn er vrijwel geheel geen veranderingen geweest sinds 1990, en die zullen er ook niet meer komen in de toekomst); en er is een enorme hoeveelheid gebruikers over de hele wereld, waarvan de meeste maar al te bereid zijn om hun kennis over te dragen aan wie die nodig heeft. Zodoende kunnen problemen die een gevolg zijn van een gebrek aan ervaring met \TeX snel worden opgelost door een bericht te sturen naar één van de \TeX -verbonden maillijsten en nieuwsgroepen (zelfs diegenen zonder netwerk-toegang zijn niet afgesloten, aangezien het TUG (\TeX Users Group) kantoor zelfs telefonische ondersteuning levert van 3 uur 's morgens tot laat in de avond - een service die *niet* alleen beschikbaar is voor leden van TUG).

Kortom, tijdens de jaren tachtig kwan \TeX boven als *het* standaard pakket voor computer-zetten: het was beschikbaar voor bijna elk beschikbaar systeem, stuurprogramma's werden geschreven voor alles van dot-matrix printers tot 2400 dpi fotozetters (maar *niet* voor margrietwiel printers!), en een steeds groeiende hoeveelheid aan publicaties verscheen die gezet waren met \TeX , danwel

T_EX als onderwerp hadden, of beide. Veel wetenschappelijke journals adopteerden T_EX (of een van de afgeleiden zoals L^AT_EX, wat gezien kan worden als een wat beperkter maar meer gebruikersvriendelijk 'front-end' voor T_EX) als het standaard formaat waarin artikelen aangeleverd moesten worden. Aangezien een auteur heel eenvoudig een proef van een artikel kon maken op zijn lokale implementatie van T_EX, en aangezien T_EX *garandeerde* dezelfde resultaten op te leveren ongeacht op welk systeem het draaide, werd het aantal contacten tussen auteur en uitgever geminimaliseerd, en iedereen had daar voordeel van. Bovendien, aangezien T_EX was ontworpen door een wiskundige, en aangezien een groot gedeelte van de *raison d'être* was geweest het mogelijk te maken om wiskunde vrijwel even eenvoudig te zetten als broodtekst, verliep de acceptatie van T_EX binnen de wiskundige gemeenschap zo mogelijk nog sneller dan binnen de algemene wetenschappelijke en academische wereld.

Om een eenvoudig voorbeeld te geven van het waarom T_EX ideaal geschikt is voor het zetten van wiskundig werk, overwegen we de hieronder staande groep vergelijkingen:

$$\begin{aligned}
 \left(\int_{-\infty}^{\infty} e^{-x^2} dx \right)^2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} dx dy \\
 &= \int_0^{2\pi} \int_0^{\infty} e^{-r^2} r dr d\theta \\
 &= \int_0^{2\pi} \left(-\frac{e^{-r^2}}{2} \Big|_{r=0}^{r=\infty} \right) d\theta \\
 &= \pi.
 \end{aligned} \tag{11}$$

Een wiskundige die dit met de hand schrijft zal vrijwel zeker beginnen met het meest linkse element van de eerste regel, om vervolgens van links naar rechts te werken, ondertussen afwisselend tussen de basisregel en de super- en subscripten als de logica dat dicteert. Een pure WYSIWYG ('What You See Is What You Get') tekstverwerker daarentegen zou van de typist eisen dat die elke rij van de formules zou verdelen in horizontale banen (dus de bovenste baan zou bijvoorbeeld alleen de tekens ∞ , 2, ∞ en ∞ bevatten), en deze dan laag voor laag in te voeren. Omdat in het algemeen WYSIWYG systemen voorgaande of volgende lijnen niet automatische aanpassen als een tussenliggende regel verkort of verlengd wordt, is het corrigeren van zulke formules extreem lastig en foutgevoelig. Nieuwere WYSIWYG-achtige systemen eisen weer een andere aanpak waarbij de auteur de formule moet invoeren in de volgorde die wordt gedicteerd door de ontleed-boom van het programma. Het hoeft niet eens vermeld te worden dat ook deze aanpak onredelijke eisen stelt aan de auteur.

T_EX staat het de wiskundige toe formules in te voeren in de meest natuurlijke manier, links beginnen en rechts eindigend; de uitlijning wordt automatisch bijgehouden als er toevoegingen of verwijderingen plaatvinden, en zelfs het horizontaal lijnen van de vier = tekens wordt automatisch uitgevoerd, vrijwel volledig onafhankelijk van de lengte van de linker- en rechter-delen. Om dit duidelijker

te maken volgt hieronder de exacte code die gebruikt werd om de tabel te zetten:

```

$$
\eqalignno
{\biggl(
  \int_{-\infty}^{\infty} e^{-x^2} \, dx
\biggl)^2
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty}
  e^{-(x^2+y^2)} \, dx \, dy \ \cr
&= \int_0^{2\pi} \int_0^{\infty}
  e^{-r^2} r \, dr \, d\theta \ \cr
&= \int_0^{2\pi}
  \biggl( -\frac{e^{-r^2}}{2} \Big|_{r=0}^{r=\infty} \biggr)
  \, d\theta \ \cr
&= \pi. \tag{11} \ \cr
}
$$

```

Het is de moeite waard om op te merken dat T_EX de spaties in de wiskundige tekst volledig negeert. Dit is omdat de regels voor het toevoegen van witruimte binnen wiskundig zetwerk ingewikkeld zijn, en niet verwacht kan worden dat die regels correct begrepen worden door een simpele wiskundige! Daardoor is de layout van de bovenstaande formules enkel en alleen voor het gemak van de auteur, en wordt volledig genegeerd door T_EX, die veel meer geïnteresseerd is in speciale tekens zoals dollartekens, backslashes, accolades, streepjes, dakjes en ampersands. En terwijl elk van deze tekens een speciale betekenis heeft voor T_EX (een dollarteken bijvoorbeeld begint en eindigt een stuk wiskundige tekst) kan toch elk van deze betekenissen eenvoudig worden aangepast, toegewezen aan een ander teken, of zelfs volledig uitgeschakeld worden indien de functionaliteit niet nodig is. Bijvoorbeeld, als een toetsenbord geen backslash kende, zou het eenvoudig zijn om de betekenis daarvan te koppelen aan een andere toets (zoals bijvoorbeeld de yen, als het een japans toetsenbord betrof).

Bovendien kunt u zien dat T_EX gemakkelijk te onthouden opdrachtreesen ('commando's', vooraf gegaan door een backslash) gebruikt. Om een paar voorbeelden te noemen: `\int` stelt een integraalteken voor, `\infty` staat voor oneindig, `\exp` de exponent-operatie (het exponentieelteken e) en zo verder. Samengestelde subscripten en superscripten worden gepresenteerd in de logische volgorde in plaats van volgens de verticale volgorde op de pagina, en de mogelijkheid is open om T_EX hints te geven over de logische structuur van de expressie, zodat (bijvoorbeeld) `\,` gebruikt wordt om wat extra wit toe te voegen voor een differentiaal zoals $d\theta$, waardoor zowel de verschijning als de leesbaarheid van de formule verbeterd wordt.

De aantrekkingskracht van T_EX voor wiskundigen is duidelijk: een hoogst logische taal voor het markeren, die ingegeven kan worden door middel van elk willekeurig toetsenbord; de mogelijkheid om een grote verzameling wiskundige symbolen te gebruiken; de uitvoer is volgens professionele standaards; ondersteuning door een grote verzameling journals; en de mogelijkheid om een

proef uit te draaien op alles vanaf een dot-matrix printer tot een 600 dpi laser printer. Voeg daaraan toe de nu alom aanwezige mogelijkheid om het resultaat te bekijken op een computerscherm (iets waarvan de eerste T_EX-promotors slechts konden dromen), en het wordt moeilijk uit te leggen waarom een willekeurige wiskundige met de beschikking over een computer T_EX *niet* voor zijn artikelen zou gebruiken!

Maar het gebruik van T_EX is niet alleen beperkt tot wiskundigen en Noord-Amerikanen, en tijdens de T_EX User Group conferentie in 1989 drong een invloedrijke en welbespraakte groep Europese T_EX gebruikers zich op aan Knuth en slaagde erin hem ervan te overtuigen dat, ondanks zijn stelling van de vorige dag dat de ontwikkeling van T_EX afgelopen was, er toch zaken ontbraken die ervoor zorgden dat T_EX geheel nutteloos was voor het grootste deel van de wereld, aangezien hoewel het zich perfect gedroeg in talen zonder accenten, T_EX vreselijk tekortschoot bij het zetten van talen waarin meer accenten voorkwamen dan slechts af en toe een enkele diacriet. Knuth, die onderkende dat het argument geldig was, gaf toe dat er iets gedaan moest worden.

Het resultaat van dit alles was T_EX 3: T_EX 82 werd bekend simpelweg T_EX 2, en T_EX 3 werd de Enige Echte T_EX. In de praktijk gebeurde dit niet zomaar: degenen die geen behoefte hadden aan de uitgebreide diacrieten-ondersteuning van T_EX 3 bleven gewoon T_EX 2 gebruiken, een een tijdlang werden T_EX macro schrijvers gedwongen hoogst *defensive* code te schrijven die eerst de omgeving moest controleren voordat er aannames gedaan konden worden over (bijvoorbeeld) het aantal verschillende tekens waarmee T_EX intern om kon gaan (dat was 128 vóór T_EX 3, en 256 daarna). Met de verschijning van T_EX 3 maakte Knuth *absoluut* duidelijk dat dit echt het einde was van de ontwikkeling van T_EX: hij had betere dingen te doen met zijn tijd, en T_EX was nu bevroren (uitgezonderd het oplossen van essentiële bugs, die hij bleef uitvoeren, maar alleen dan als aangetoond kon worden dat het repareren essentieel was). Bovendien maakte hij het even duidelijk dat T_EX *niet* verder ontwikkeld mocht worden door iemand anders: hij wilde T_EX als *zijn* schepping overlaten voor zijn nageslacht, tot in alle eeuwigheid, en niet als *zijn*-schepping-zoals-aangepast-door-iemand-anders.

In het algemeen werd dit goed opgevat door de T_EX wereld: Knuth staat in enorm hoog aanzien bij diegenen die T_EX gebruiken, en er waren er slechts heel weinigen die het idee voorstonden om zijn wensen te negeren en bereid waren veranderingen aan T_EX voor te stellen. Maar er was een redelijk grote groep T_EX gebruikers, waaronder de schrijver dezes, die van mening waren dat als T_EX *niet* veranderde, het eenvoudig zou sterven. Niet omdat er fundamentele gebreken in T_EX waren – algemeen wordt aanvaard dat dat er daarvan slechts heel weinig zijn – maar omdat de wereld niet heeft stilgezeten sinds 1978, en hoewel een script-taal toen state-of-the-art geweest mag zijn, is dat zeker nu niet meer het geval. Daarbij komt dat hoewel het aantal interne tekens was verhoogd van 128 naar 256, Knuth maar heel weinig tot niets had gedaan aan

ondersteuning van *asiatische* talen, waarin het aantal verschillende tekens moet worden uitgedrukt in duizenden of tienduizenden. En als laatste waren er mensen die ervan overtuigd waren dat op *sommige* gebieden flinke vooruitgang kon worden geboekt (vooral vanuit het standpunt van de macro programmeur, ook bekend als de 'format writer' als het pakket macros een compleet functioneel systeem betreft) met een relatief kleine investering met betrekking tot het veranderen van T_EX.

De uitvoering van deze ideeën is waarschijnlijk de speerpunt van de T_EX technologie vandaag de dag: bedrijven zoals Blue Sky hebben ogenblikkelijke/toenemende T_EX interpreters geproduceerd, die in staat zijn om veranderingen in de broncode van een T_EX document direct te laten zien; Advent Publishing maakten 3B2, dat zowel een grafische als een tekstuele specificatie van de layout toelaat, waarbij de veranderingen in de één automatisch worden doorgevoerd in de ander; John Plaice en Yannis Haramboulos hebben een 64-bits versie van T_EX ontwikkeld die intern gebruik maakt van unicode; en de groep waarmee ik het meest verbonden ben (de NTS groep, waarbij NTS staat voor 'New Typesetting System') heeft een volledig compatibele opvolger van T_EX gemaakt met de naam e-T_EX, die functionaliteit toevoegt zonder de compatibiliteit in gevaar te brengen (de NTS groep heeft ook het voornemen T_EX te her-implementeren vanaf niets in één van de moderne snelle-typing talen zoals Prolog of CLOS. Het idee is het toestaan van snelle experimenten met alternatieve zet-algoritmes of paradigma's). Of één van deze ideeën zal aanslaan valt nog te bezien, hoewel de Apple Macintosh *aficionados* Classic Textures (het eerder genoemde produkt van Blue Sky) erg hoog aanslaan. Een fundamentele vraag is die van stabiliteit: aangezien een van de belangrijkste sterke punten van T_EX zijn stabiliteit is, kan alleen de toekomst uitwijzen wat de wereld zal vinden van een systeem dat specifiek bedoeld is om te blijven reageren en evolueren, in plaats van star en versteend te zijn.

Wat misschien de moeite van het vermelden waard is, is dat al deze projecten ervoor gezorgd hebben dat Knuth's wensen gehonoreerd werden, niet alleen naar de letter maar ook naar de geest: niet één ervan streeft ernaar zichzelf T_EX te noemen (Het project van John Plaice en Yannis Haramboulos noemt zich zelfs Omega, wat absoluut nooit verward kan worden met T_EX), terwijl ze toch allemaal toegeven veel verschuldigd te zijn aan Knuth en T_EX: zonder hen zou geen van deze andere projecten ooit het licht van de dag gezien hebben.

Parallele Ontwikkelingen

Natuurlijk stond de rest van de wereld niet stil terwijl T_EX aan het evolueren was: de computerwetenschap bleef zichzelf ontwikkelen, en computer-netwerken verspreidden zich van het laboratorium naar het leger en de universiteiten en uiteindelijk naar de hele wereld. Regel-georiënteerde editors vielen in onmin en werden vervangen door volledig-scherm editors (behalve in

de wat achterlopende wereld van MS/DOS, die tot relatief kortgeleden alleen EDLIN bleef aanbieden). Scriptgeoriënteerde markup talen zoals de ROFF familie die eerder genoemd werd werden aangevallen door steeds slimmer wordende tekstverwerkers, en WYSIWYG ('What You See Is What You Get'), GUI ('Graphical User Interface') en WIMP ('Windows, Icons, Menus and Pull-down lists') werden de orde van de dag.

Rond dezelfde tijd dat Knuth begon met het werk aan \TeX , her-implementeerden John Warnock en Martin Newell, werkzaam bij Xerox PARC, een eerdere taal ('the Design System') als JaM ('John and Martin!'). Vanuit dit geclusterde begin ontwikkelden zich uiteindelijk zowel de Interpress (Xerox print protocol) als de PostScript taal. Terwijl Interpress relatief onbekend bleef, sloeg Adobe PostScript in als een bom: voor de eerste keer was er een *de facto* pagina-beschrijvingstaal, die het toestond ingewikkelde pagina's te beschrijven als een algoritme (en dus erg efficiënt). Hoewel Hewlett Packard's Printer Control Language (PCL) wijd ondersteund en nageemaakt bleef (en is), vestigde PostScript zichzelf als *de* standaard voor hoge-kwaliteit printers (waarmee ik laser-printers en beter bedoel), en al vrij snel streefden printer-producerende bedrijven ernaar om PostScript interpreters of PostScript emulators te leveren voor hun high-end producten. Helaas (voor de schrijvers van de emulators) is PostScript een complexe taal, en veel van de eerdere emulators waren gebrekig op één of meer manieren. Adobe, zijnde de ontwikkelaars van de taal, hadden natuurlijk veel minder last van deze problemen, maar zelfs zij maakten verbeterde versies van hun interpreter beschikbaar naarmate de tijd vorderde.

Gedurende een hele tijd bleven gedeelten van PostScript een goed bewaard geheim: de mysterieuze `eexec` operator was niet gedocumenteerd, en hoewel het PostScript handboek informatie gaf over het formaat van de zogenaamde 'Type 3' fonts, bleven de net zo mysterieuze 'Type 1' fonts ongedocumenteerd. Natuurlijk is reverse engineering een goed begrepen gereedschap en eindelijk werden de grenzen doorbroken: beschrijvingen van `eexec` begonnen op te duiken in de pers en ten lange leste liet Adobe zich vermurwen en stond de volledige documentatie van zowel `eexec` als hun Type 1 fonts af.

Niet veel later vestigden Type 1 fonts zich net zo als de standaard voor fonts als PostScript dat was voor pagina-beschrijvingstalen; bedrijven zoals Corel begonnen hun eigen Type 1 fonts te produceren, die nauw aansloten bij de industriële standaardfonts, maar die net genoeg afweken (op zijn minst qua naam) om beschuldigingen van fontpiraterij te kunnen ontlopen (hoewel dit laatste probleem tot aan de dag van vandaag een zorg blijft voor de top van de font ontwerpers, zoals Hermann Zapf). Alle belangrijke font-gieterijen begonnen hun fonts aan te bieden in Type 1 formaat, en velen beloofden *al* hun fonts binnen afzienbare tijd beschikbaar te hebben in Type 1 formaat. De zogenaamde 'font magic' die het vroege Adobe fonts toestond redelijke resultaten op te leveren op lage-resolutie apparaten zoals 300 dpi laserprinters werd hernoemd naar 'font hinting', en ook dit werd uiteindelijk

gedocumenteerd door Adobe. Nieuwe mogelijkheden bleven toegevoegd worden aan de PostScript taal, en in 1990 kondigde Adobe een volledig nieuwe versie van de taal aan, 'PostScript Level 2'. Deze nieuwe versie bevatte alle eerdere toevoegingen aan de taal, en introduceerde tevens vele nieuwe mogelijkheden zoals de mogelijkheid om compacte (binaire) representaties van een PostScript document te gebruiken, zowel als de eerdere (ASCII) representatie; nieuwe kleur-modellen werden toegevoegd, en ondersteuning werd beschikbaar voor composiet-fonts.

PostScript was oorspronkelijk ontstaan als een interne taal voor printers, maar het werd al snel duidelijk dat een versie van PostScript die in staat was een computer-scherm te besturen extreem nuttig zou zijn. Adobe creëerde een eigen versie hiervan met de naam 'Display PostScript', maar ondertussen was L. Peter Deutsch al begonnen met het werk aan een eigen PostScript interpreter met de naam 'GhostScript', en een van de fundamentele functionele eigenschappen hiervan was dat het in staat was om het scherm te besturen van elke computer waarop het gebruikt werd (het bevatte ook een ruime verzameling stuurprogramma's voor non-PostScript printers, en pseudo-stuurprogramma's voor een paar van de meer populaire grafische uitwisselformaten). Gedurende 1995 kondigde Peter eindelijk GhostScript versie 3 aan, welke een vrijwel complete PostScript Level 2 emulatie besloeg. En hoewel de officiële Adobe interpreter een gelicenseerd (en relatief duur) produkt bleef, was en bleef GhostScript een gratis programma voor diegenen die het niet gebruiken om er geld mee te verdienen. De computerwereld is een grote dank verschuldigd aan L. Peter Deutsch, zowel voor zijn kunde in het schrijven van GhostScript als voor zijn vrijgevigheid in het gratis beschikbaar stellen van het programma, en ook is grote dank verschuldigd aan al de verschillende personen die hun privé stuurprogramma's en/of verbeteringen aan het GhostScript project hebben gedoneerd (PS-View van Bogusław Jackowski en Piotr Pirowski verdient extra vermelding).

Van het ARPAnet naar het Web

Een paar jaar voordat Knuth begon aan het werk aan \TeX , was het Amerikaanse leger in de personificatie van [D]ARPA (het [Defence] Advanced Research Projects Agency) een piloot-project begonnen met het doel computers met elkaar te verbinden over heel lange afstanden. Hoewel verbindingen tussen lokale computers niet ongevoelbaar waren, waren verbindingen over duizenden kilometers toen nog onbekend, maar [D]ARPA onderkende de potentiële militaire belangrijkheid van zulke verbindingen en startte daarom een hele serie onderzoeksprojecten met als doelstelling zulke verbindingen mogelijk te maken. Deze projecten begonnen oorspronkelijk geïsoleerd van elkaar, maar zodra het test-netwerk beschikbaar was kreeg het project een voorwaartse duw – feitelijk een eigen bestaan – uit zichzelf, en de gehele ontwikkelings-strategie vanaf dat moment vond plaats door discussie *via* zowel als *over* het netwerk. Dit netwerk, dat om voor-de-hand-liggende rede-

lyx: een front-end voor L^AT_EX of een Textprocessor?

Taco Hoekwater

Samenvatting

Dit artikel is een recentie van het programma LyX. LyX is een programma voor X windows dat dient als (bijna-)wysiwyg schil bovenop L^AT_EX. De huidige stabiele beta-versie, die hier dan ook besproken wordt, is versie 0.10.7. Het programma wordt geschreven door een aantal auteurs onder ‘leiding’ van Matthias Ettrich (ettrich@informatik.uni-tuebingen.de).

1 Introductie

Zoals we allemaal heel wel weten, is T_EX weliswaar een uitstekende typesetter, maar de manier van omgaan met de verschillende programma's is voorzichtig uitgedrukt enigszins ouderwets. Aan de andere kant zijn tekstverwerkers weliswaar prettig in het gebruik, maar is de uitvoer vaak om te huilen.

Programma's die op een structurele manier met text omgaan, in staat zijn redelijke uitvoer te leveren bij zwaar wiskundige artikelen én gebruikersvriendelijk zijn zijn met de vingers van één hand te tellen. Ook de commerciële software laat op dit gebied te wensen over, eigenlijk is er alleen maar Scientific Word, wat alleen beschikbaar is voor Microsoft Windows, niet voor de vele mensen die Unix gebruiken.

Het is dan ook een moeilijke taak om een manier te vinden waarop de kwaliteiten van (bijvoorbeeld) T_EX behouden blijven binnen een grafische omgeving. Het programma dat hier besproken worden, LyX, doet een poging hiertoe. LyX is een Unix programma dat een “what you see is what you mean” schermopbouw aanbiedt voor L^AT_EX bestanden.

Uit de man pagina:

LyX is a modern approach of writing documents with a computer which breaks with the tradition of the obsolete typewriter concept. It is designed for people who wants a professional output with a minimum of time effort, without becoming specialists in typesetting. Compared to common word processors, LyX will increase the productivity a lot, since most of the typesetting will be done by the computer, not the author. With LyX the author can concentrate on the contents of his writing, since the computer will take care of the look.

Let wel: LyX is alleen een grafische schil. Het zorgt ervoor dat wat op het scherm te zien valt vertaalt wordt naar tekst die voor L^AT_EX begrijpelijk is.

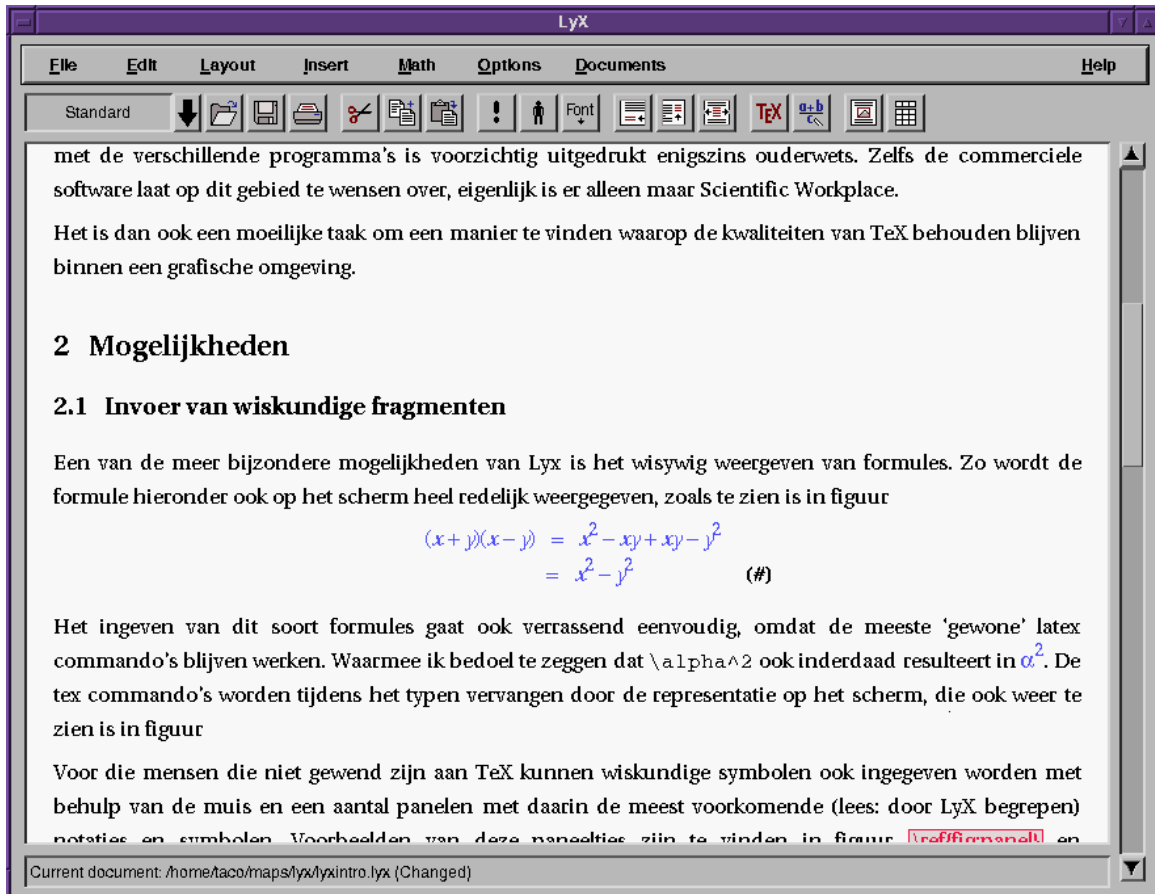
LyX zelf kan alleen wat simpele formattering op het scherm uitvoeren, en is zeker geen typesetter.

1.1 Bestandsformaat

LyX gebruikt een eigen bestandsformaat om tekst in te bewaren. Hoewel het voor de gemiddelde T_EX gebruiker redelijk herkenbaar is (het heeft wel wat van Eijckhout's systeem weg), is het toch heel duidelijk geen T_EX:

```
#This file was created by
<taco> Tue Apr 1 13:27:28
1997
#LyX 0.10 (C) 1995 1996
Matthias Ettrich and the LyX
Team
\lyxformat 2.10
\textclass article
\language default
\inputencoding default
\fontscheme times
\epsfig dvips
\papersize a4paper
\paperfontsize 11
\baselinestretch 1.00
\secnumdepth 2
\tocdepth 2
\paragraph_separation skip
\quotes_language english
\quotes_times 1
\paperorientation portrait
\papercolumns 2
\papersides 1
\paperpagestyle headings

\layout Title
\fill_top \fill_bottom
```



Figuur 1: Hoofdscherm van LyX

```
lyx: een front-end voor LATEX
of een Textprocessor?
\layout Author
```

```
Taco Hoekwater
\layout Abstract
```

```
Dit artikel is een recentie
van het programma lyx.
Lyx is een programma voor X
windows dat dient als
(bijna-)wysiwyg schil
om latex.
De huidige stabiele versie,
die hier dan ook besproken
wordt, is versie
0.
10.
7
\layout Standard
```

waarde T_EX tegeven, en deze tekst wordt dan rechtstreeks doorgegeven aan T_EX, inclusief eventuele commando's die erin staan. Eventuele tekortkomingen aan de interface worden hierdoor opgevangen.

1.2 Invoer

Configuratiebestanden vertellen het programma de samenhang tussen de schermuitvoer en de gebruikte L^AT_EX class. Daardoor is het via de grafische interface alleen mogelijk die omgevingen in te voeren die in het bijbehorende bestand zijn gedefinieerd.

Invoer is mogelijk met behulp van de muis (knoppen en menu's), maar ook via het toetsenbord. De 'commando's' van LyX zijn met behulp van een opstartscript te verbinden met toetsen. Twee bestanden met 'standaard' definities zijn bijgesloten bij de LyX distributie, en deze zorgen naar keuze voor bindingen die lijken op emacs of de CUI-style (selectie via shift-pijltje). Voor het invoeren van accenten ondersteunt LyX 'dode toetsen'.

Conversie tussen dit formaat en L^AT_EX code gebeurt door een intern filter. Gelukkig is dit filter behoorlijk slim. Zo is het mogelijk om tekst als karakter-style de

1.3 Invoer van wiskundige fragmenten

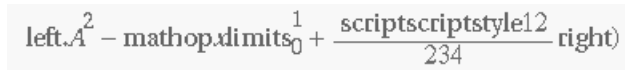
Een van de mogelijkheden van Lyx is het wysiwyg weergeven van formules. Zo wordt de formule hieronder ook op het scherm heel redelijk weergegeven, zoals te zien is in figuur 1.

$$\begin{aligned} (x + y)(x - y) &= x^2 - xy + xy - y^2 \\ &= x^2 - y^2 \end{aligned} \quad (1)$$

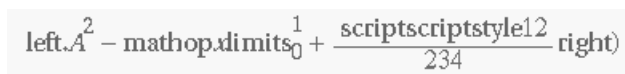
Het ingeven van formules gaat ook verrassend eenvoudig, omdat de meeste ‘gewone’ latex commando’s blijven werken in de ‘wiskundige editor’ van lyx. Binnen lyx zijn er twee sterk van elkaar verschillende mogelijkheden voor het invoeren van wiskunde, waarbij de ene methode alleen geschikt is voor mensen die al ervaring hebben met L^AT_EX, terwijl de andere methode juist geschikt is voor mensen die in het geheel niets van latex afweten.

De eerste methode is gebaseerd op ‘interpretatie terwijl u typt’. U kunt overgaan naar de wiskundige modus van lyx door te klikken op het bijbehorende icoon, of door een toetscombinatie zoals (alt-M f). Na het overschakelen naar de wiskundige modus wordt de letterlijke toetsenbordinvoer `\alpha^2` ook correct geïnterpreteerd en resulteert in α^2 . De tex commando’s worden tijdens het typen vervangen door de representatie op het scherm, die ook weer te zien is in figuur 1.1. Deze vorm van herkenning gaat behoorlijk ver, zo is het bijvoorbeeld mogelijk om een formule achteraf nog even aan te passen door het toevoegen van coderingen. Verreweg het belangrijkste gegeven hier is echter dat alles dat niet wordt begrepen rechtstreeks wordt doorgegeven aan T_EX. Dat maakt formules zoals de volgende mogelijk:

$A^2 - x_0^1 + \frac{12}{234}$ wordt op het scherm weergegeven als:



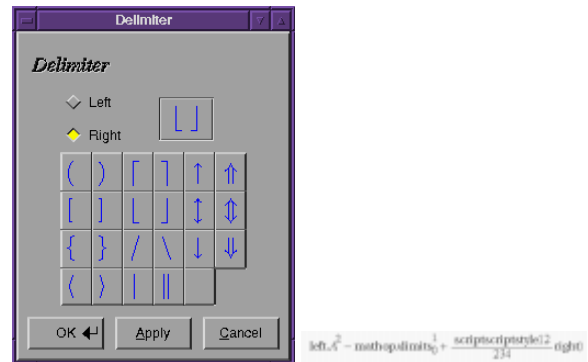
Op het scherm worden de herkende commando’s weergegeven in blauw, terwijl de codes zoals `mathop` zijn weergegeven in rood.



Figuur 2: Het wiskundige hoofdpaneel

Voor die mensen die niet gewend zijn aan T_EX kunnen wiskundige symbolen ook ingegeven worden met behulp van de muis en een aantal panelen met daarin de

meest voorkomende (lees: door LyX begrepen) notaties en symbolen. Voorbeelden van deze paneeltjes zijn te vinden in figuur 2 en figuur 3.



Figuur 3: Wiskundige sub-panels

Voorlopig beslaan de panelen slechts een subverzameling van de commando’s die herkend worden door T_EX zelf, maar toch is de beschikbare set al behoorlijk omvangrijk, en valt er zonder meer mee te werken voor een wetenschappelijk artikel van ‘gemiddelde’ zwaarte.

1.4 Tabellen

Aangezien heel LyX gebaseerd is op L^AT_EX, zal het nauwelijks verbazen dat alle tabellen worden vertaald naar de `tabular` omgeving. Voorlopig zijn de mogelijkheden voor het maken van tabellen nogal beperkt, wat niet wegneemt dat De volgende tabel mogelijk bleek (let op de multicolumn entry):

Kind	Oranges	Pears	Other fruit	
People	1	1234	4	0
cats, dogs and other animals	234	0	23	12

Ook hier geldt weer dat het geen enkel probleem is om over te gaan op standaard T_EX mogelijkheden, zoals ik inderdaad moest doen voor ‘cats , dogs and other animals’, maar elegant is het natuurlijk niet.

1.5 Figuren

Ondersteuning voor figuren is ronduit uitstekend. PostScript figuren worden weergegeven op het scherm in de correcte grootte, en alle ‘standaard’ opties van `psfig` zijn eenvoudig aan te klikken. De bijbehorende dialoog is weergegeven in de figuur hieronder (4), een voorbeeld van de representatie op het scherm in figuur 5.

scherm worden de herkende commando's weergegeven in blauw, terwijl de codes zoals `mathop` zijn weergegeven in rood.

Voor die mensen die niet gewend zijn aan TeX kunnen wiskundige symbolen ook ingegeven worden met behulp van de muis en een aantal panelen met daarin de meest voorkomende (lees: door LyX begrepen) notaties en symbolen. Voorbeelden van deze paneeltjes zijn te vinden in figuur. [\ref{fig:panel}](#) en figuur. [\ref{fig:subpanel}](#)

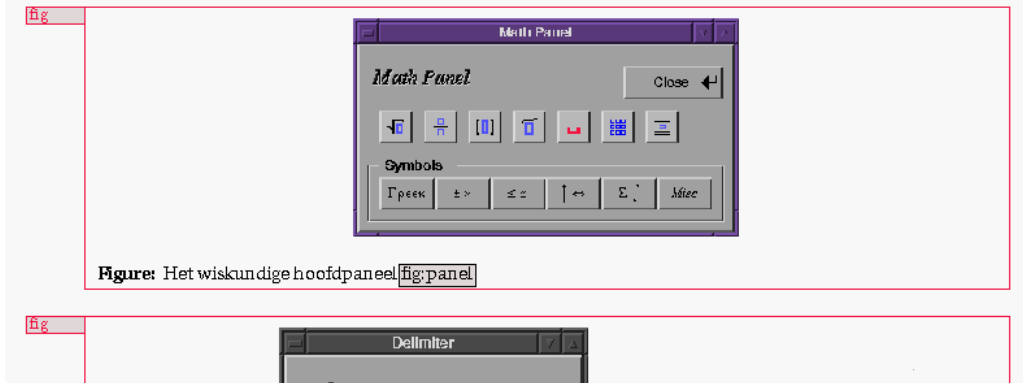
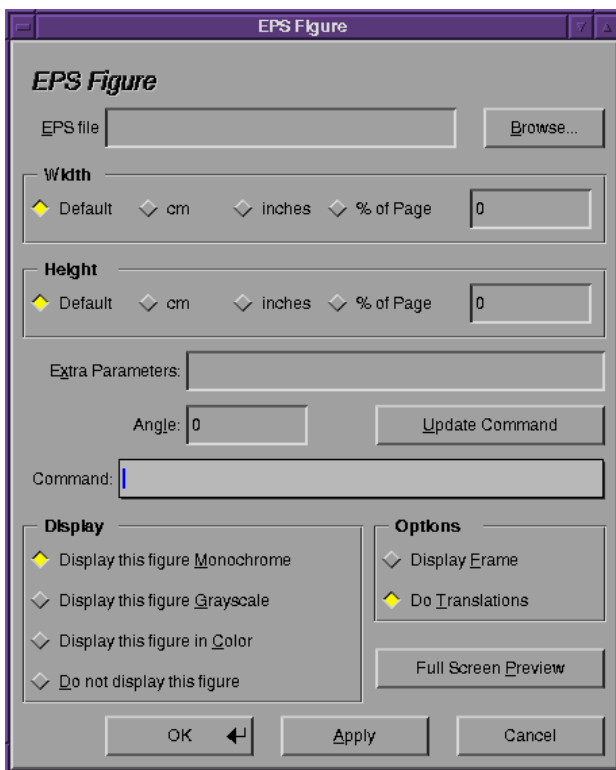


Figure: Het wiskundige hoofdpaneel [fig:panel](#)

Figuur 5: on-screen display van figuren



Figuur 4: Het configuratiescherm voor invoegen van figuren

1.6 Kruisreferenties

Het referentiesysteem is eenvoudig maar afdoende. Een nieuw label wordt toegevoegd via de menu keuze 'Insert Label', terwijl 'Insert Cross-reference' ervoor

zorgt dat het vrijwel onmogelijk is geworden om typfouten te maken¹. Een popup laat de mogelijke referenties zien, die vervolgens alleen nog maar aangeklikt hoeven te worden. Er kan dus alleen gerefereerd worden aan al bestaande labels.

1.7 Diversen

Een paar andere dingen die in LyX aanwezig zijn zijn onder andere:

- **Spellingscontrole.** De spellingscontrole is gebouwd als een front-end voor `ispell`. Het correct werken van de spellingscontrole is dus afhankelijk van of `ispell` correct geconfigureerd is, maar het voordeel van deze aanpak is wel dat er voor vrijwel elke taal wel een woordenboek te vinden is. Het is mogelijk om alleen spellingscontrole uit te voeren op de geselecteerde tekst, waardoor het mogelijk is om meerdere talen te gebruiken binnen een document. Er moet dan wel een nieuwe `ispell` gestart worden.
- **Table of Contents.** LyX houdt een inhoudsopgave bij in het geheugen. Deze is te openen als een apart window, dat gebruikt kan worden voor navigatie door het document. Vooral met wat grotere artikelen is dat heel prettig: gewoon klikken op de sectie waar je heen wilt.
- **De 'primaire selectie' van de X server kan geplakt worden als regels of paragrafen.**

¹Ook hier is het weer mogelijk om direct `\labels` en `\refs` in te tikken via de TeX escapes, maar zeker in dit geval is dat een slecht idee. Inconsistenties zijn vrijwel onmogelijk zolang de aangeboden structuur wordt gebruikt..

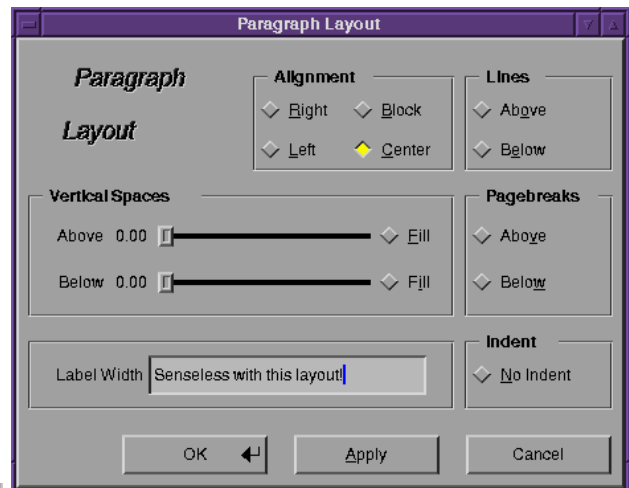
- Via het File menu is het echter eenvoudig om L^AT_EX het document te laten formatteren, xdvi te starten of te printen via bijvoorbeeld dvips.
- Undo en redo. Beide zijn 'oneindig'.
- Voetnoten en margenoten. Alle typen floats kunnen op het scherm 'gesloten' worden, zodat er alleen nog een klein woordje overblijft dat het type aangeeft.
- Het is mogelijk om hfills, protected space en linebreaks in te voeren. Daardoor is het mogelijk om rechts uitgevulde regels te maken. In de figuur hieronder is te zien dat dit op het scherm correct wordt aangegeven.



- De style van paragrafen kan worden aangepast met behulp van een dialoog. Hierdoor is het mogelijk om typen paragrafen te gebruiken waarvoor geen 'standaard' style beschikbaar is. Met name is dit nodig voor het maken van raggedright paragrafen, maar er zijn ook andere trucs mogelijk, zoals te zien is in de dialoog (weergegeven in figuur 6).
- Op soortgelijke wijze is het mogelijk om de style van karakters / woorden aan te passen. Er wordt geen exact font gedefinieerd, maar de verschillende onderdelen van de NFSS beschrijving van het font kunnen per stuk worden aangepast.
- Het is mogelijk om na het draaien van L^AT_EX direct naar de volgende fout (indien aanwezig natuurlijk) te springen binnen het document. Dubbelklikken op de aanduiding [Error] in het bestand, laat vervolgens de foutmelding van T_EX zien:



Uiteraard is dat vrijwel altijd een 'undefined control sequence', die is ontstaan door het incorrect met de hand ingeven van een T_EX commando.



Figuur 6: De paragraaf-style dialoog.

- Export van L^AT_EX bestanden is mogelijk. Hoewel deze bestanden beginnen met de definitie van een aantal L^YX-eigen omgevingen en commando's, zijn er geen aparte stylefiles nodig om de gegenereerde bestanden te compileren. De overdraagbaarheid van de bestanden is dus uitstekend.²
- L^YX kan meerdere documenten tegelijk openen, hoewel er slechts 1 window beschikbaar is. Wisselen tussen verschillende documenten gebeurt via het 'Documents' menu.
- Direct printen is mogelijk door 1 druk op de knop. L^YX zorgt ervoor dat L^AT_EX vaak genoeg wordt aangeroepen om te zorgen dat de referenties correct zijn. Printen gebeurt via dvips.
- De geheugeneisen van L^YX vallen heel erg mee, een bestand van 2 Megabyte (800 pagina's tekst) neemt slechts ongeveer 4 Megabyte geheugen in beslag, en laadt in ongeveer 40 seconden op mijn P75. Compilatie van dat bestand duurde ruim vijf minuten.
- Linuxdoc SGML wordt ondersteund als documentclass. Daardoor is het buitengewoon eenvoudig geworden om SGML bestanden te maken (en van daaruit natuurlijk weer info, html etc.). Het is wel noodzakelijk hiervoor Linuxdoc/sgmltools geïnstalleerd te hebben.

2 Tekortkomingen

Natuurlijk is een project zoals L^YX een enorme onderneming, en ontbreekt er nog van alles. De low-

²Dit is een van de punten waar L^YX het duidelijk wint van Scientific Workplace. Een figuur in L^YX is gewoon een 'figure' omgeving met een \epsfig regel erin.

level mogelijkheden van editen in emacs zijn natuurlijk nooit haalbaar binnen een wysiwyg systeem, maar toch zijn er nog wel een paar dingen die node ontbreken.

- Import en export filters. De huidige situatie is dat LyX alleen bestanden kan lezen in eigen formaat en als ascii, terwijl er alleen geexporteerd kan worden naar L^AT_EX. Hoog op mijn verlanglijstje staat in ieder geval import van Rich Text Format.
- Subdocumenten. LyX gaat er van uit dat het volledige document wordt bewaard in 1 bestand. Hoewel er geen dwingende reden is om een bestand te willen splitsen vanwege de interface, is dit vooral bij grote projecten een ernstig gemis.
- Betere zoek-ervang mogelijkheden. Zoek/ervang kan tot op heden alleen als letterlijke tekst en 1 voor 1. Het is bijvoorbeeld niet mogelijk om te zoeken op paragraaf-stylen.
- Aan het invoegen van met name tabellen valt nog het een en ander te verbeteren. In de huidige stand van zaken zijn alleen eenvoudige tabellen in te voegen.
- Ook het invoeren van met name de wat vreemdere wiskundige constructies zou verbeterd moeten worden. Onder andere ontbreekt op dit moment ondersteuning voor diagrammen, en is het niet mogelijk om symbolen te definiëren (zoals ‘lim inf’). Elke formule is een L^AT_EX eqnarray ‘of een plain display, hetgeen lang niet altijd gewenst is. Natuurlijk kan met de hand een en ander geregeld worden, maar uiteraard is dat niet de bedoeling van een mooie grafische schil.
- Het is jammer dat de scherm-layout niet geprint kan worden in plaats van de dvips uitvoer. Vooral voor korte briefjes zou dat erg handig zijn. Voor korte memo’s is een tekstverwerker nu eenmaal handiger dan L^AT_EX, en LyX doet toch al een groot gedeelte van het werk dat je van een tekstverwerker zou verwachten.
- Er valt nog wel wat toe te voegen aan de L^AT_EX foutcontrole, zoals controle op overfull en underfull boxes.
- De documentatie loopt achter op de ontwikkelingen van de programmeurs. Hoewel dat normaal is voor ‘beta’ programma’s, is het te hopen dat de kwaliteit van de docs snel verbetert. LyX is al met al een complex systeem, voor mensen die iets willen aanpassen aan het standaard gedrag is correcte documentatie dan ook absoluut noodzakelijk.

Het hardst ontbreken echter interfaces naar L^AT_EX stylefiles. Alleen de ‘standaard’ L^AT_EX classes en amsart worden ondersteund, waardoor het maken van een document vaak een twee-traps raket wordt: eerst de tekst invoeren in LyX, een L^AT_EX bestand exporteren, en vervolgens dat bestand aanpassen zodat het conformeert aan het gewenste stylefile. Deze situatie is verre van ideaal, omdat het resulterende bestand niet opnieuw ingelezen kan worden in LyX.

Het is wel mogelijk om definitie-bestanden te maken voor andere stylefiles, maar de documentatie van LyX is op dit punt nog niet echt in orde, zodat het aanpassen van zo’n bestand min of meer ‘blind’ gebeurt. Een klein stukje uit zo’n definitie-bestand volgt hieronder:

```
# General textclass parameters
Columns                1
Sides                  1
PageStyle              Headers
MaxCounter             Counter_Section
DefaultFamily          Roman
DefaultSeries          Medium
DefaultShape           Up
DefaultSize            Normal

# Standard style definition
Style Standard
  Margin                Static
  LatexType
Paragraph
  LatexName             dummy
  ParIndent             MM
  ParSkip               0.4
  Align                 Block
  AlignPossible        Block,
Left, Right, Center
  LabelType
No_Label
End

# Section-numbered style
definition
```

3 Conclusie

Samenvattend komt het er op neer dat LyX nog lang niet klaar is, maar niettemin al uitstekend geschikt voor het verwerken van eenvoudige artikelen en brieven. Gezien de snelheid waarmee het project zichzelf ontwikkelt, ziet de toekomst er veelbelovend

uit. Wellicht het belangrijkste aan L_YX is dat het niet beweert iets anders te zijn dan een schil om L^AT_EX. Voor gebruikers die (bijna) niets van T_EX weten, valt er eenvoudig mee te werken, en ook voor T_EXperts is het een prima hulpmiddel tijdens het intypen van broodtekst.

kdvi: een verbeterde xdvi

Taco Hoekwater

4 april 1997

Samenvatting

kdvi is een aangepaste versie van xdvi, een DVI previewer voor het X Window Systeem. De belangrijkste verandering is het gebruik van een 'echte' grafische interface ter vervanging van de nogal primitieve widgets van xdvi. kdvi is *alpha* software, de huidige versie is 0.2. De auteur is Markku Hihnala (mah@ee.oulu.fi).

Keywords: DVI previewer, X, interface

1 Intro

Al heel lang is er een (wat kinderlijke) discussie gaande tussen Microsoft Windows gebruikers en Unix-gebruikers. Het betreft dan de relatieve kwaliteiten van de op deze platformen beschikbare dvi previewers, respectievelijk dviwin en xdvi. kdvi is een nieuw programma dat de Windows gebruikers nogal wat wind uit de zeilen neemt door het drastisch verbeteren van de xdvi interface.

2 De voordelen van xdvi

De sterke punten van xdvi tegenover andere previewers waren (en zijn):

- Automatische generatie van font-bitmaps. Zowel METAFONT als PostScript fonts kunnen dynamisch aangemaakt worden.
- Het ondersteunen van virtuele fonts, iets wat dviwin bijvoorbeeld nog altijd niet doet.
- On-screen PostScript. xdvi laat niet de eventuele bitmap previews in Encapsulated PostScript zien, maar biedt (vrijwel) volledige ondersteuning voor PostScript figuren én specials via GhostScript.
- Hoge kwaliteit van de display door een uitzonderlijk snelle en goede anti-aliasing routine.
- Snel zoeken van font bitmap bestanden door gebruik van de kpathsea bibliotheek.

3 De nadelen van xdvi

Anderzijds is er 1 enorm nadeel aan xdvi: de interface met de gebruiker is op z'n best primitief te noemen. Ik heb wel eens horen beweren dat xdvi het schoolvoorbeeld is

van hoe je de interface van een X programma *niet* moet programmeren.

Figuur 1 laat een typisch voorbeeld zien van een xdvi scherm. Opvallend is (behalve dan dat de fonts er goed uitzien), dat de knoppen ronduit primitief zijn. Er is zelfs niet een hint van een bitmap, laat staan 3-D buttons. Er is ook geen menubalk en zelfs geen on-line help.

Erg ergerlijk, maar niet te zien in de figuur is dat er ook nog (!) knoppen buiten het window staan aan de onderkant van het scherm. Het window is te klein om ervoor te zorgen dat die knoppen zichtbaar worden.¹

4 Even wat achtergrond: Het KDE project

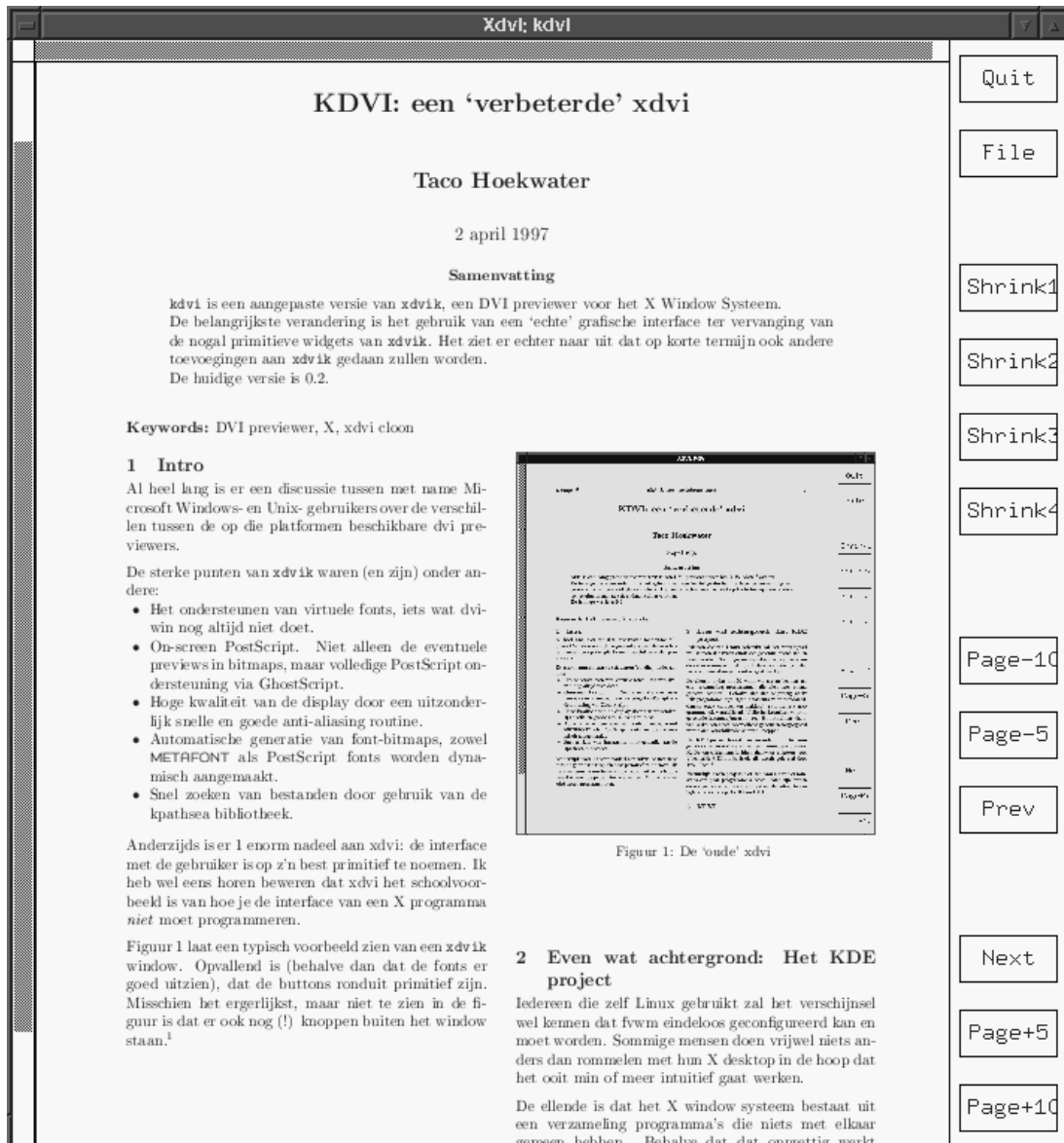
Iedereen die zelf Linux gebruikt zal het verschijnsel wel kennen dat het X Window Systeem eindeloos geconfigureerd kan en moet worden. Sommige mensen doen vrijwel niets anders dan rommelen met hun X desktop in de hoop dat het ooit min of meer intuïtief gaat werken.

Het probleem is dat het X window systeem bestaat uit losse programma's die feitelijk niets met elkaar gemeen hebben. Behalve dat dat onprettig werkt (elk programma zijn eigen shortcuts voor standaard-dingen zoals knippen en plakken), gebruikt elk programma ook verschillende bibliotheekroutines voor de getoonde knoppen/menu's etc. Het resultaat daarvan is dat een grote hoeveelheid RAM geheugen weggegooid wordt aan verschillende soorten knoppen.

Het KDE project [1] hoopt hier verandering in te brengen door het introduceren van een heuse desktop voor X. De ontwikkelaars hebben daartoe gekozen voor 1 bepaalde GUI bibliotheek, die wordt geleverd door Troll Tech [2].²

¹Mijn X display is 1100x850, toch niet echt klein. Ik heb werkelijk geen idee hoe ik die knoppen in beeld moet zien te krijgen zonder een virtuele desktop te installeren.

²Deze C++ bibliotheek is vrij te gebruiken voor vrij verspreide software, is beschikbaar in zowel een binaire als een source distributie, is heel makkelijk in het gebruik, en identiek aan het commerciële product van Troll Tech. Dat laatste zorgt voor een uitstekende kwaliteitsbewaking



Figuur 1: De 'oude' xdvi

Natuurlijk is zo'n project niet haalbaar als er niet minstens een paar programma's beschikbaar zijn buiten de systeemprogramma's, en zie daar de geboorte van kghostview, ktop, kedit en kdvi.

5 KDVI

Figuur 2 laat duidelijk het verschil in interface zien tussen xdvi en kdvi. Al de verschillende onderdelen van de interface zoals de menu's en de scrollbars kunnen ook onzichtbaar gemaakt worden indien gewenst, om ruimte te maken voor een groter gedeelte van de DVI file.

Er is een menubalk, met menu's voor File (Open, Close, Print), View (magnify, demagnify), Page (next, previous, Goto), Options (show PS, make PK's, en opties voor het verbergen of laten zien van verschillende onderdelen van de GUI).

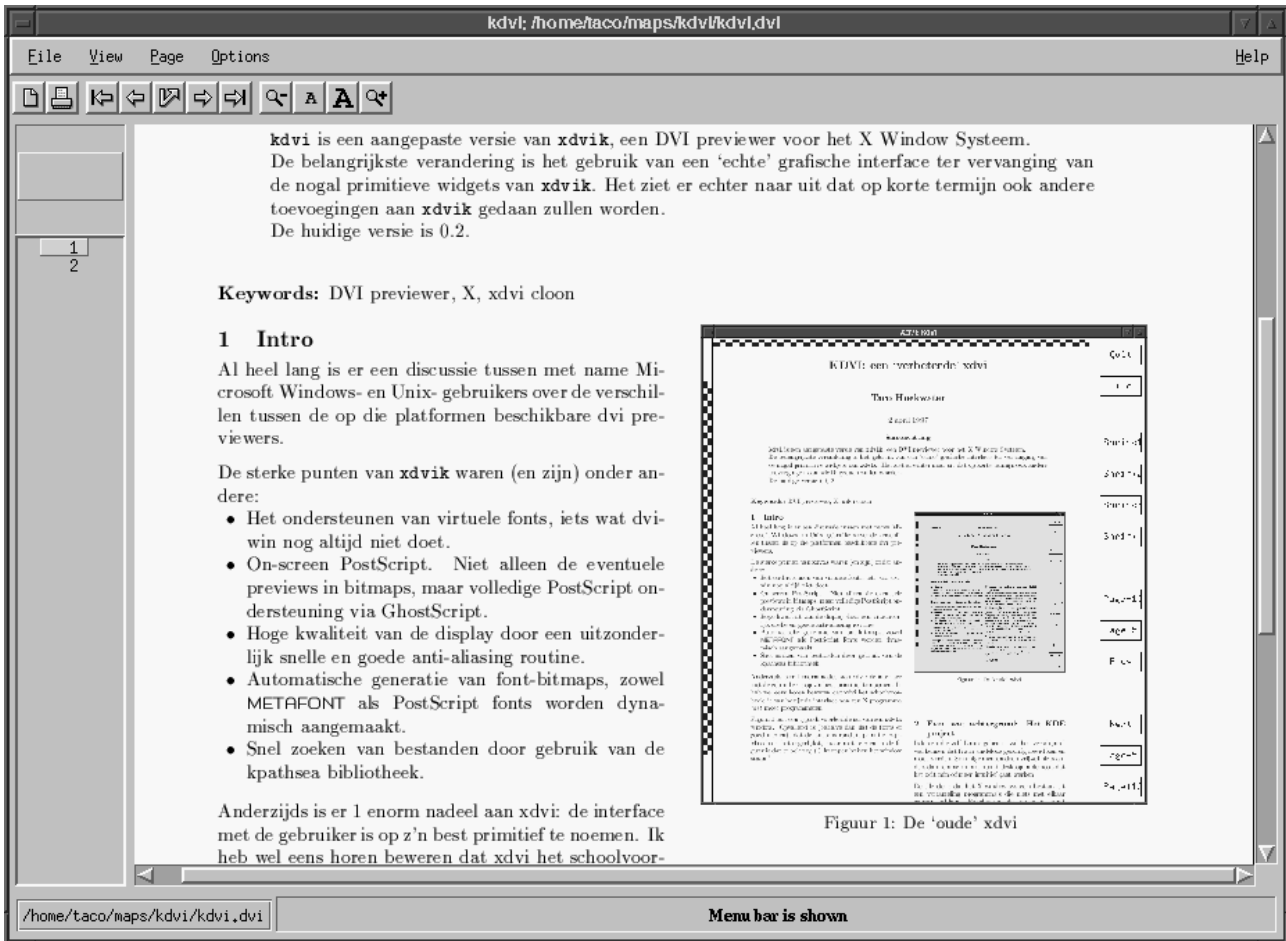
Daaronder een knoppenbalk, met ruwweg dezelfde functie als de menu's. De menu's en de knoppenbalk kunnen onafhankelijk van elkaar verborgen worden.

Het merkwaardige dingetje rechtsboven is een scrollwidgeet. Je kunt dat heen en weer slepen over de pagina om een ander gedeelte te zien.

Daaronder een paginalijst. Het is mogelijk via deze lijst snel naar een andere pagina te springen (als de lijst te lang wordt verschijnt er uiteraard een scrollbar). Tevens is het mogelijk pagina's te markeren voor printen.

Dan is er nog de statusbalk onderin het scherm, die op het moment nogal kaal is, en nog niet echt functioneel. En als laatste de scrollbars langs de getoonde DVI pagina.

De linker muisknop sleept het getoonde gedeelte van de pagina's heen en weer, net als in acrobat's reader.



Figuur 2: kdvi

De rechter muisknop bevat een context-menu.

Alle commando's zijn ook op de ouderwetse manier in te geven met een toetsbord-commando, maar de shortcuts zijn niet allemaal hetzelfde als in xdvi, en bovendien nog aan verandering onderhevig.

6 Toevoegingen van kdvi

kdvi bestaat in de kern uit de code van xdvi 18f, met een schil eromheen die voor extra functionaliteit zorgt.

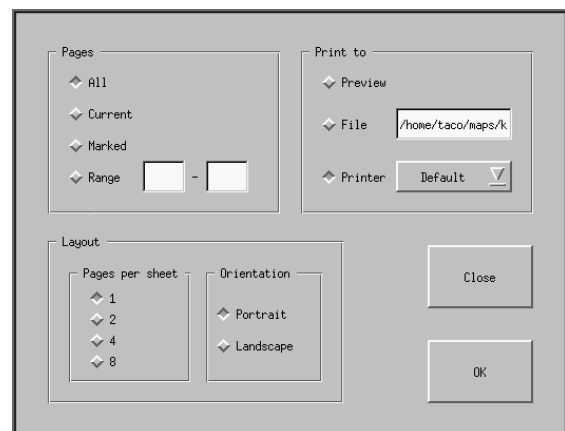
6.1 Gemarkeerde pagina's

Het is mogelijk verschillende pagina's te markeren via de paginalijst links. Er is bijvoorbeeld de mogelijkheid om alle even of alle oneven pagina's te selecteren, maar een voor een kan ook. Voorlopig is het grootste nut hiervan de mogelijkheid om op een simpele manier een gedeelte van een DVI te printen.

6.2 Printen

Een toevoeging van kdvi is een print dialoog. Het is mogelijk om via kdvi een dvi bestand te printen met dvips zonder verdere noodzaak voor commandoregel-

opdrachten. Er zijn zelfs wat extra mogelijkheden zoals 2-up printen (het programma mpage is noodzakelijk om te zorgen dat dit correct werkt), en het printen van gemarkeerde pagina's.



Figuur 3: De print dialoog

6.3 De muis

Een ander verschil met `xdvik` is het gebruik van context-menu's onder de muisknop. Door het ingenieuze scroll-widget is er niet meer zo'n noodzaak voor het hebben van een vergrootglas.

6.4 Configuratie

De configuratie van `kdvi` is duidelijk anders dan die van `xdvik`. Alle programma's van de KDE maken gebruik van een tekstbestand met eenzelfde syntax als `win.ini` voor de configuratie, in plaats van de standaard `application default files` van X.

Een van de voordelen van deze aanpak is dat `kdvi` zijn schermopbouw onthoudt voor de volgende keer (niet alleen de opties, maar ook de lokatie en grootte van de windows, en een lijst van pas geleden geopende bestanden).

Uiteindelijk zal volledige configuratie (ook van de gebruikte kleuren) mogelijk zijn via de menu's, dit gedeelte van de code was echter nog niet af in de door mij geteste versie.

6.5 Drag en Drop

`Kdvi` ondersteunt het droppen van DVI bestanden via de KDE filemanager, `kfm`. Doordat de manier waarop de verschillende programma's met elkaar praten 'KDE standaard' is, zal dat niet werken bij het gebruik van `kdvi` zonder de KDE.

7 Wat er nog ontbreekt

Uit het 'todo' lijstje van de auteur:

- Media selection, currently a4 size is hard coded. (This only affects the borders drawn)
- More options
- Better online help (no coding just writing)
- Better print dialog
- More KDE keyboard commands with `kkeyconfig`
- Enhance the GUI
 - option dialogs
 - better status line
 - nice icons
 - KDE Standard Look (?)
- Redirect error messages to a window (partly done)
- D&D support (drops accepted now)
- Better PostScript handling (currently not very robust)
 - antialiasing
- Improved building and installation
- More background processing to keep the UI responsive

Anything else? Feel free to make suggestions and report any problems you have.

Markku Hihnala mah@ee.oulu.fi

8 Problemen

Het versienummer zegt het al, eigenlijk: `kdvi` is alpha/beta software. Dat betekent dat een en ander niet altijd even vlekkeloos werkt.

In de praktijk valt er uitstekend met het programma te werken, maar met name is enige voorzichtigheid geboden bij het kiezen van grote vergrotingen. De cache die ervoor zorgt dat je moeiteloos over de hele pagina heen kan schuiven gebruikt soms wel heel erg veel geheugen.

Bij DVI files waarbij de eerste pagina's anders genummerd zijn dan de rest van het bestand (boeken bijvoorbeeld) is het niet goed mogelijk om alleen de geselecteerde pagina's te printen. Dat verschijnsel is een gevolg van de commandoregel van `dvips`, en is ook zonder het gebruik van `kdvi` wel eens een probleem.

Het programma verlangt nogal wat andere bestanden om correct te functioneren, waardoor de drempel voor niet-kde gebruikers helaas behoorlijk groot wordt. Op zijn minst zijn noodzakelijk:

- De Qt dynamic library van Troll Tech
- De KDE dynamic libraries
- Een C++ compiler, want het programma wordt alleen gedistribueerd als source code.
- Een correct geconfigureerde \TeX distributie (bij voorkeur $\text{te}\TeX$). Deze is noodzakelijk voor het vinden/plaatsen van `pk` bestanden.³
- Een executable die `xdvi` heet en in het pad staat⁴

Verder zijn de volgende programma's noodzakelijk voor bepaalde gedeeltes van de functionaliteit:

- Het programma `kdehelp` om te zorgen dat het Help menu werkt. De documentatie is trouwens in HTML, dus het kan ook met alleen een web-browser.
- GhostScript voor in-line PostScript preview.
- `dvips` voor het printen.
- `mpage` voor 2-up printen.
- De KDE file manager (`kfm`) als u graag dvi files wilt kunnen droppen.
- Ghostview (`gv`) is beschikbaar als 'printer'.

Referenties

- [1] Het KDE project is te bereiken via <http://www.kde.org>
- [2] Troll Tech Inc. zit op <http://www.troll.no>

³Een statisch gecompileerde versie van de laatste versie voor Linux ELF systemen is beschikbaar via de NTG.

⁴Dit is een nogal bizarre bug.

DVIPS-manual—good old portability, and some more

Kees van der Laan

Abstract

Portable manuals should be written and marked up in this way. Just a few macros to take care of the formatting. The manual can be easily personalized. The dvips manual has been made available as a BLUe report with an index—based on IR-s by Rokicki—and a ToC. Both are welcome when using the manual. Some experience in using dvips under UNIX is also reported: overprinting a picture, adding a caption, and aligning pictures. A pitfall in using `\psfig` is mentioned as well.

Keywords: Aligning figures, BLUe format, captions, dvi, eps, figures, minimal markup, overprinting, portability, PostScript.

1 Introduction

This note is about my experience with the dvips manual, with respect to the markup and the contents.

POSTSCRIPT can be seen as a no-nonsense, refreshing approach to printing, well ... to describe what should be printed. Fundamental was to embrace outlines, reducing complexity by an order, as an alternative to bitmaps, especially in the beginning of its introduction. Equally unusual was to explore the stack idea to its deepest. Underlying was the awareness that for computer graphics only a few primitives is all that is really needed, apart from add-ons and bells-and-whistles. The attitude to consider fonts as a special case of graphics contributed further to a uniform approach.

In short POSTSCRIPT had—and still has—much to offer to the world.

Looking back after ≈ 10 years since its introduction as POSTSCRIPT we can only conclude that POSTSCRIPT made it.¹ It is a de facto standard. Its promises have been recognized and picked up by Original Equipment Manufacturers, OEMs for short. It has been embraced as THE universal laser printer language, abstracting from the technologies of the various printers.

The software industry responded with applications which can automatically generate (E)PS, to be visualised by printers or screens, or included in (La)T_EX documents.

Tomas Rokicki in dvips—and other persons with drivers I'm not familiar with—paid the T_EX community a great service in bridging the gap between dvi files and ps files. They provided 'convertors,' to transform a dvi file into POSTSCRIPT format. His dvips not only converts but

also allows to merge POSTSCRIPT,² well ... encapsulated POSTSCRIPT, EPS for short.

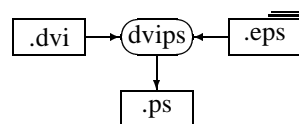


Fig.1 dvips use

dvips allows for several ways of including pictures.

1.1 Pictures interspersed with text

At the moment I use mostly

```
$$\psfig{file=...}$$
```

1.2 Pictures as background for a page

Hooks like at each beginning or ending of a page

```
/bop-hook {...}def%to be user-defined
```

are provided as well, to facilitate for example overprinting of a page.

1.3 Audience

I hope that my discussions will draw the attention of the unwearry, those who still struggle with the unusualities of (La)T_EX, that there should be a right balance between T_EXpertise, METAFONTry, and the emerging cooperating tools which are provided by the world outside.

1.4 Beyond PostScript

But, ... as time moved on limitations of POSTSCRIPT became apparent³ giving birth to successors. Maybe Adobe's

¹For the history of the predecessors see the introduction in Adobe's red book.

²The important aspect for including EPS graphics with your (La)T_EX document. I read that `\psfig` is obsolete and replaced by `\epsfig`. Hmm, not yet available on the system I'm using remotely for this purpose, at least I could not find it.

³Well, relatively spoken newer tools responded better to the growing demands and changing needs, of course.

PDF will be accepted as the successor of POSTSCRIPT, but it is too early to tell.⁴

1.5 Is the manual still needed?

Since the emerge of user interfaces to POSTSCRIPT, such as PStricks, one can ask whether a user needs this dvips manual at all. For the L^AT_EX users this seems to be true; they are served by a UI in L^AT_EX jargon, enhancing portability of the L^AT_EX script with the lower level `\special-s` taken care off for each environment. Carlisle⁵ has put it as follows.

‘At a given site the graphics packages will be customized to use a suitably back end file that converts the L^AT_EX syntax into the form required by the local driver.’

So plain T_EXies are more and more suffering from the law of diminishing returns,⁶ ... or so it seems.

For the options of the invocation of dvips one still needs the manual, or the prompt from the system when using dvips without arguments.

2 DVIPS manual

Only recently I could lay hands on the dvips manual,⁷ and I was surprised that just ≈ 50 lines of macros was all that was needed on top of plain T_EX to format the manual. Portability in the spirit of Knuth.⁸

So, the manual was easily printed. Before I collected my remotely produced prints I read the source file as T_EXies do. Ah, indeed very close to *minimal* markup. Great joy.

This is how portable manuals should be written. Just a few macros on top of plain T_EX, is all that is needed to mark up your script.

To really appreciate the contents the T_EX formatted output was more convenient. Paging through the 50 odd pages, the price which has been paid to portability became evident: no table of contents nor an index.⁹ I was puzzled.

On second thoughts, I could easily add my own bells-and-whistles and you can probably add yours. That is portability! To refrain from (too-)detailedness, to ease personalization.

2.1 Customization

Personally, I like a TOC and an Index. Very convenient for reusing the material.

2.1.1 What about a ToC?

To provide a ToC is not hard especially when printed at the end of the job. I like simple, one-pass jobs.

2.1.2 What about an Index?

The general attitude is that sorting should be done outside of T_EX, and that entails the answer. However, since my BLUe’s format, indexing can be done within T_EX on-the-fly and in one-pass.

So I decided to edit the markup of the manual. To let the IR-s conform with Knuth’s markup for IR-s, which has been adopted in BLUe’s format, to keep up compatibility with Knuth.

3 DVIPS manual as a BLUe’s report

Rokicki has used `\sec` and `\sub`, which come functionally close to a chapter heading and a subheading.¹⁰ The IR-s markup had to be adapted too. The adaptations read

```
\^{...}      into   \{...}
\^{\\tt...} into   \!...!
```

Ipsa facto for the silent variants, that is with the hats doubled.

The script was surrounded by commands as follows.¹¹

```
\input blue.tex \loadindexmacros\report
\beginscript
<script proper>
\sortindex\pasteupindex\pasteuptoc
\endscript
```

The nasty `\hoffset` and `\magnification` have been commented out among others.

The `\author` token variable was assigned a `\btable` with appropriate data.

Et voilà, the dvips manual was for reuse personalized as a BLUe’s report.

⁴PDF compensates for the size of POSTSCRIPT files, allows multi-media usage, and is suited for WWW browsing. On the other hand Erik Frambach communicated that printing of a pdf file goes via converting it first to POSTSCRIPT, for example. Printing such a file on a Laserjet III failed as yet, however. In general PDF drivers are needed.

⁵MAPS 96.2.

⁶In Dutch: ‘De wet van de remmende voorsprong.’

⁷Thank you Taco Hoekwater for providing me with the files.

⁸Knuth writes his macros on ‘demand.’ For the book Concrete Mathematics manmac as such was not used. A new set of macros was written to serve whatever was the purpose. For his TAOCP he develops a new bunch.

⁹Some sort of Index Reminders were in the file but gobbled away.

¹⁰A few chapter titles had to be added, though, demonstrating that even such knowledgeable people like Rokicki are led astray when not supported by a ToC.

¹¹Note that `\loadindexmacros` and `\report` don’t commute. When you commute them you will get the wrong size of the head font. This inelegance could have been circumvented by not giving the option for loading the index macros, but just include them as part of the kernel. This consideration might be a reminder for the right hierarchy of specification order.

4 Some experience

Functionalities of POSTSCRIPT↔(La)T_EX use—L^AT_EX biased—have been reported by Reckdahl in among others MAPS 96.2. It seems that plain T_EXies suffer from lagging behind, because it is tied up with L^AT_EX. Not true. Happily, plain T_EXies hardly need this elaboration. All what is needed is to wrap the picture in a (v)box and manipulate the box as usual, whatever the contents. From this point of view alignment and overprinting is nothing new.

4.1 Alignment of pictures

As said nothing new. How to wrap up an eps figure in a box is something different however, but independent from placement within context.¹² For the unwearied a little elaboration follows below.

Example (*Aligning pictures at the bottom*)

```
\setbox0=\vbox to 5cm{<picture1>}
\setbox1=\vbox to 2cm{<picture2>}
$$\copy0\quad\copy1$$
```

Do you see how to align at the top?

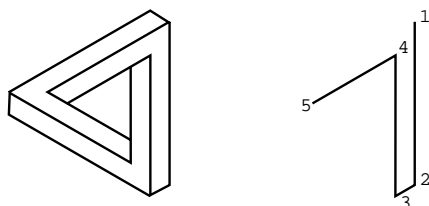
Example (*Centering pictures*)

This goes straightforwardly as follows.

```
\setbox0=\vbox to 5cm{<picture1>}
\setbox1=\vbox to 2cm{<picture2>}
$$\vcenter{\copy0}\quad\vcenter{\copy1}$$
```

Example (*Aligning EPS figures from files*)

The Escher pictures



are obtained via¹³

```
$$\psfig{file=eschera.eps,height=1in}
\psfig{file=escherb.eps,height=1in}$$
```

4.2 Captions

A similar application is the use of a caption tied up with a figure.¹⁴

The idea is the same. Instead of backskipping and overprinting—\llap—a user can box the picture together with the caption.

Example (*Caption tied up with a figure*)

The figure and caption on the first page of this note was obtained as follows.¹⁵

```
\setbox0=\hbox{\dvipspic}
$$\vbox{\copy0\vskip5pt
\hbox to\wd0{\hss
Fig.1 dvips use\hss}
}$$
```

4.3 Overprinting

Rokicki in his dvips manual provided in ‘Other bells and whistles’ an example of how to use /bop-hook for overprinting each page by a light grey, rotated and magnified representation of the word DRAFT.

In his ‘NEAT’ example he shows how to realize the functionality of a grey background with a (white) shadow. This comes down to overprint the \hrule set in some greyscale, by the word NEAT.

A related application is to overprint a greyscale picture. The principle reads as follows.

```
\setbox0=\vbox to5cm{<picture>}
\setbox1=\vbox to5cm{<overprint>}
\hbox{\copy0\hskip-\wd0\copy1}
```

In case the picture is just EPS code the contents of box0 can be built as follows, where it is assumed that the POSTSCRIPT picture is constructed above and right from the reference point. Note the sequence order of the \vss and the \special.

```
\hrule width...height0pt%to warrant space
\vss
\special{"...}
```

Details have to be synchronized. The height of the vbox must correspond with the upperright corner of the BoundingBox, if the lower left corner is the origin.¹⁶ The clipping boundary must also match these dimensions. Error-prone? Well, ...yes, but adopt a discipline from it, and know what you are doing.

¹²See the Overprint figure below as an example.

¹³For the contents of the EPS files consult my earlier note ‘Just a little bit of POSTSCRIPT’, MAPS 96.2.

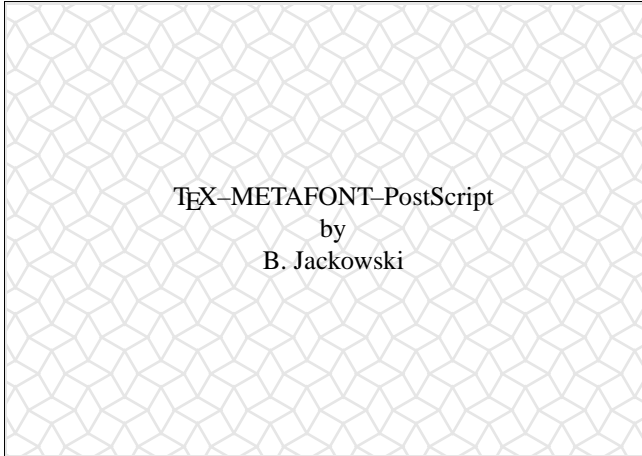
¹⁴So far, I have hardly used captions because the figures are part of examples which have the argument of the \example tag as label, or caption. It is a bit overdone to provide also for a caption.

¹⁵It looks low-level, but ... it is not so low. These commands can be easily wrapped into a macro. The point is that the few lines make explicit what one is doing, in a nutshell. The advantages are that one does not have to learn new things, and BLUE—Ben Lee User—remains in control.

¹⁶1 inch corresponds to 72 units of POSTSCRIPT.

I'm sorry that the example is not as simple as could have been, because I chose for an intriguing background.¹⁷

Example (*Overprint a (greyscale) picture*)



via

```
{\parindent0pt\medskip\hsize8.47cm %\vsize25cm
\def\ok{\hbox{\boxit{\vbox to6cm
  {\hrule width\hsize height0pt
\sss\special{"
/a 10 def
/tena {a 10 mul}def
/ha {a .5 mul} def
/tile {ha 3 sqrt mul 0 moveto
  0 ha lineto
  ha 3 sqrt mul neg 0 lineto
  0 ha neg lineto
  closepath
  ha 1 3 sqrt add mul ha 3 sqrt mul lineto
  ha 2 3 sqrt add mul 0 lineto
  ha 1 3 sqrt add mul ha 3 sqrt mul neg lineto
  closepath
} def
/frame {0 0 moveto
  0 170 rlineto
  240 0 rlineto
  0 -170 rlineto
  closepath
} def
/dotiling{
7{gsave
  11{tile a 1 3 sqrt add mul 0 translate
  }repeat stroke
  grestore
  gsave ha 1 3 sqrt add mul dup translate
  11{tile a 1 3 sqrt add mul 0 translate
  }repeat stroke grestore
  0 a 1 3 sqrt add mul translate
}repeat
} def
%tile
```

```
gsave
frame gsave stroke grestore clip
a neg dup translate
0.9 setgray dotiling
grestore}%end \special
}}
%
\llap{\vbox to6cm{\vfil
\centerline{\strut\TeX--METAFONT--PostScript}
\centerline{\strut by}
\centerline{\strut B. Jackowski}
\vfil}%end vbox
}}}%end ok
%
\ok
\smallskip}
```

Explanation. The (overprinted) vboxes are wrapped up in a hbox to prevent a (line-)break between the vbox-es. The boxit command is there to show the boundary on my Mac where the special is not executed.

Remarks. Note the dependencies. Size of the vbox-es, and the size of the frame, each takes its own units, cm versus bp. Do you see the variant to frame it by an oval?

4.4 A pitfall in using psfig

Inclusion of eps files in BLUE scripts worked. But, ...it hiccuped with my transparencies. It turned out that reading of the BoundingBox data assumes that the e-o-l is changed (by T_EX) into a space. My transparencies format is line-oriented which entails that the e-o-l is changed into a \par. The following mod in the straight invocation was necessary.

```
{\catcode'\^=7 \catcode'\
  $$\psfig{...}$$}
```

5 Conclusions

The dvips manual which comes with version 5.58 is highly portable and a genuine example of minimal markup. In general it is an example of how a manual should be done to enhance portability.

It lacks a ToC and Index, however. The latter facilitate reading and browsing. Grace to the few macros and the IR-s markup already present, these elements can be easily added. Slight adaptations could compensate for these, at the expense of being dependent on BLUE's format \report and sorting within T_EX.

The next step is probably to provide a representation of the manual as a hypertext.

The use of dvips—\special-s—is not so portable as yet.¹⁸ As long as there is no generally accepted \special

¹⁷In the tutorial of Adobe's blue book a simple example of overprinting within PostScript alone is provided on p.42, about the business card for the Diamond Cafe. Here we are combining the best of both worlds: PostScript for the graphics and T_EX for the formatting, merging PostScript parts within a T_EX script.

¹⁸Rokicki works on a standard for \special-s. Keep fingers crossed.

syntax—a standard—I’ll do as much graphics as I can by T_EX alone via

- stacking of boxes,
- using of gkppic macros of Knuth (the picture functionality), or
- my turtle graphics macros.

These facilities come with BLUe’s format and have been documented in the Publishing with T_EX user’s guide. With respect to my POSTSCRIPT pictures I concentrate on merging—just including the eps file—keeping the interface as thin as possible.

It is demonstrated further that alignment of figures is nothing special when the figures are wrapped up in boxes. How to create the contents of those boxes is another story.

Related to Reckdahl’s note, MAPS 96.2, it’s amazing what a little insight can do.

Finally, when an error is made in the POSTSCRIPT code I was frustrated by the fact that the printer did not give me feedback, read, there were no pages from the erroneous page onwards.

Have fun, and all the best.

Appendix: ToC of dvips

Although the page numbers won’t correspond with the original manual the structure of the manual can be read from the following ToC.

1	dvips	
1.1	Why Use dvips?	1
1.2	Using dvips	2
1.3	Paper Size and Landscape Mode	2
2	Including PostScript Graphics	
2.1	The Bounding Box Comment	3
2.2	Using the EPSF Macros	4
2.3	Header Files	5
2.4	Literal PostScript	5
2.5	Literal Headers	6
2.6	Other Graphics Support	6
2.7	Dynamic Creation of PS Graphics Files	8

3	Using PostScript Fonts	
3.1	The afm2tfm Program	9
3.2	Changing a Font’s Encoding	11
3.3	Special Effects	12
3.4	Non-Resident PostScript Fonts	13
3.5	Font Aliases	14
4	Font Naming Conventions	
4.1	Foundry	16
4.2	Typeface Families	16
4.3	Weight	16
4.4	Variants	17
4.5	Expansion	17
4.6	Naming Virtual Fonts	17
4.7	Examples	18
5	Command Line Options	
6	Configuration file	
6.1	Configuration File Searching	23
6.2	Configuration File Options	23
7	Automatic Font Generation	
8	Path Interpretation	
9	Environment Variables	
10	Other Bells And Whistles	
11	MS-DOS and OS/2	
12	Installation	
13	Diagnosing Problems	
13.1	Debug Options	34
13.3	No Output At All	34
13.3	Output Too Small or Inverted	34
13.4	Error Messages From Printer	35
13.5	400 DPI Is Used Instead Of 300 DPI	35
13.6	Long Documents Fail To Print	35
13.7	Including Graphics Fails	35
13.8	Can’t Find Font Files	35
13.9	Can’t Generate Fonts	36
14	Using Color with dvips	
14.1	The Macro Files	37
14.2	User Definable Colors	38
14.3	Subtleties in Using Color	38
14.4	Printing in Black/White, after Colorizing	39
14.5	Configuring dvips for Color Devices	39
14.6	Protecting Regions From Spurious Colors	39
14.7	Color Support Details	40
14.8	Index	42
15	Table of Contents	

How to handle compound and bounded words

just another hyphenation method

Hans Hagen

April 1 1996

One of T_EX's strong points in building paragraphs is the way hyphenations are handled. Although for real good hyphenation of non-english languages some extensions to the program are needed, fairly good results can be reached with the standard mechanisms and an additional macro, at least in Dutch.

1 \unprotect

ConT_EXt originates in the wish to typeset educational materials, especially in a technical environment. In production oriented environments, a lot of compound words are used. Because the Dutch language poses no limits on combining words, we often favor putting dashes between those words, because it facilitates reading, at least for those who are not that accustomed to it.

In T_EX compound words, separated by a hyphen, are not hyphenated at all. In spite of the multiple pass paragraph typesetting this can lead to parts of words sticking into the margin. The solution lays in saying `spoelwater||terugwinunit` instead of `spoelwater-terugwinunit`. By using a one character command like `|`, delimited by the same character `|`, we get ourselves both a decent visualization (in T_EXedit and colored verbatim we color these commands yellow) and an efficient way of combining words.

The sequence `||` simply leads to two words connected by a hyphen. Because we want to distinguish such a hyphen from the one inserted when T_EX hyphenates a word, we use a bit longer one.

`spoelwater||terugwinunit` `spoel- wa- ter- te- rug- win- unit` spoelwater-terugwinunit

As we already said, the `|` is a command. This commands accepts an optional argument before it's delimiter, which is also a `|`.

`polymeer|*|chemie` `po- |y- meer* che- mie` polymeer*chemie

Arguments like `*` are not interpreted and inserted directly, in contrary to arguments like:

`polymeer|~|chemie` `po- |y- meers che- mie` polymeerchemie
`|(|polymeer|)|chemie` `(po- |y- meer-) che- mie` (polymeer)chemie
`polymeer|(|chemie)|` `po- |y- meer (-che- mie-)` polymeer(chemie)

Although such situations seldom occur —we typeset thousands of pages before we encountered one that forced us to enhance this mechanism— we also have to take care of comma's.

`op||, in|| en uitstellen` `op- in- en uit stel len` op-, in- en uitstellen

The next special case (concerning quotes) was brought to my attention by Piet Tutelaers, one of the driving forces behind rebuilding hyphenation patterns for the dutch language.¹ We'll also take care of this case.

`AOW|'|er` `AOW- en` AOW'er
`cd|'|tje` `cd- tje` cd'tje
`ex|-|PTT|'|er` `ex- PTT- en` ex-PTT'er
`rock|-|'n|-|roller` `rock- 'n- roller` rock-'n-roller

The mechanism described here is one of the older inner parts of ConT_EXt. The most recent extensions concerns some special cases as well as the possibility to install other characters as delimiters. The preferred way of specifying compound words is using `||`, which is installed by:

`\installdiscretionaries || -`

Some alternative definitions are:

¹In 1996 the spelling of the dutch language has been slightly reformed which made this topic actual again.

```

\installdiscretionaries ** -
\installdiscretionaries ++ -
\installdiscretionaries // -
\installdiscretionaries ~~ -

```

after which we can say:

```

test**test**test   test test test   test-test-test
test++test++test   test test test   test-test-test
test//test//test   test test test   test-test-test
test~~test~~test   test test test   test-test-test

```

Now let's go to the macros. First we define some variables. In the main ConTeXt modules these can be tuned by a setup command. Watch the (maybe) better looking compound hyphen.

```

2 \def\compoundhyphen    {-}\kern-.25ex{-}
\def\beginofsubsentence {---}
\def\endofsubsentence   {---}

```

The last two variables are needed for subsentences —like this one— which we did not yet mention.

We want to enable breaking but at the same time don't want compound characters like - or – to be separated from the words. T_EX hackers will recognise the next two macro's:

```

3 \def\prewordbreak  {\penalty10000\hskip0pt\relax}
\def\postwordbreak  {\penalty0\prewordbreak}

```

We first show the original implementation, which only supports | as command and delimiter. Before activating | we save it's value:

```

\edef\domathmodediscretionary{\string|}

```

after which we're ready to define it's meaning to:

```

\catcode'\|= \@@active

\protected\def|{%
  {\ifmmode
    \expandafter\domathmodediscretionary
  \else
    \expandafter\dotextmodediscretionary
  \fi}

```

We need a two stage \futurelet because we want to look ahead for both the compound character definition and the (optional) comma that follows it, and because we want to prevent that T_EX puts this comma on the next line. We use \next for easy and fast checking of the argument, we save this argument (which can consist of more tokens) and also save the character following the |#1| in \nextnext.

```

\def\dotextmodediscretionary%
  {\bgroup
   \futurelet\next\dodotextmodediscretionary}

\def\dodotextmodediscretionary#1|%
  {\def\betweendiscretionaries{#1}%
   \futurelet\nextnext\dododotextmodediscretionary}

```

The main macro consists of quite some \ifx tests while \checkafterdiscretionary handles the commas. We show the simplified version here:

```

\def\dododotextmodediscretionary%
  {\let\nextnextnext=\egroup
   \ifx | \next
     \checkafterdiscretionary
     \prewordbreak\hbox{\compoundhyphen\nextnext}\postwordbreak
   \else\ifx=\next
     \prewordbreak\compoundhyphen
   \else\ifx~\next

```

```

\discretionary{-}{\thinspace}\postwordbreak
\else\ifx(\next
\prewordbreak\discretionary}{(-){\prewordbreak
\else\ifx)\next
\prewordbreak\discretionary{-)}{)}\postwordbreak
\else\ifx'\next
\prewordbreak\discretionary{-}{}'\postwordbreak
\else
\checkafterdiscretionary
\prewordbreak\hbox{\betweendiscretionaries\nextnext}\postwordbreak
\fi\fi\fi\fi\fi\fi
\nextnextnext}

```

```

\def\checkafterdiscretionary%
{\ifx,\nextnext
\def\nextnextnext{afterassignment\egroup\let\next=}%
\else
\let\nextnext=\relax
\fi}

```

The most recent implementation is more advanced. As demonstrated we can install delimiters, like:

```
\installdiscretionaries || \compoundhyphen
```

This time we have to use a bit more clever way of saving the math mode specification of the character we're going to make active. We also save the user supplied compound hyphen. We show the a bit more traditional implementation first.

```

\def\installdiscretionaries#1%
{\catcode'#1\@@other
\expandafter\doinstalldiscretionaries\string#1}

\def\doinstalldiscretionaries#1%
{\setvalue{mathmodediscretionary#1}{#1}%
\catcode'#1\@@active
\dodoinstalldiscretionaries}

\def\dodoinstalldiscretionaries#1#2%
{\setvalue{textmodediscretionary\string#1}{#2}%
\protected\def#1{\discretionarycommand#1}}

```

A bit more *catcode* and character trickery enables us to discard the two intermediate steps. This trick originates on page 394 of the *T_EXbook*, in the appendix full of dirty tricks. The second argument has now become redundant, but I decided to reserve it for future use. At least it remembers us of the symmetry.

```

4 \def\installdiscretionaries#1#2#3%
  {\setvalue{mathmodediscretionary\string#1}{\char'#1}%
  \setvalue{textmodediscretionary\string#1}{#3}%
  \catcode'#1=\@@active
  \scratchcounter=\the\uccode'~
  \uccode'~='#1
  \uppercase{\protected\def~{\discretionarycommand~}}%
  \uccode'~=\scratchcounter}

5 \def\dohandlemathmodebar#1%
  {\getvalue{mathmodediscretionary\string#1}}

6 \def\discretionarycommand%
  {\ifmmode
  \expandafter\dohandlemathmodebar
  \else
  \expandafter\dotextmodediscretionary
  \fi}

```

Although adapting character codes and making characters active can interfere with other features of macropackages, normally there should be no problems with things like:


```
\installdiscretionary || +
\installdiscretionary ++ =
```

The real work is done by the next set of macros. We have to use a double `\futurelet` because we have to take following characters into account.

```
7 \def\dottextmodediscretionary#1%
  {\bgroup
  \def\dodottextmodediscretionary##1#1%
    {\def\betweendiscretionary{##1}%
    \futurelet\nextnext\dododottextmodediscretionary}%
  \let\discretionarycommand=#1%
  \def\textmodediscretionary{\getvalue{textmodediscretionary}\string#1}%
  \futurelet\next\dodottextmodediscretionary}

8 \def\dododottextmodediscretionary%
  {\let\nextnextnext=\egroup
  \ifx\discretionarycommand\next
    \checkafterdiscretionary
    \prewordbreak\hbox{\textmodediscretionary\nextnext}\postwordbreak
  \else\ifx=\next
    \prewordbreak\textmodediscretionary
  \else\ifx~\next
    \prewordbreak\discretionary{-}{\thinspace}\postwordbreak
  \else\ifx(\next
    \ifdim\lastskip>\!zeropoint\relax
      (\prewordbreak
      \else
        \prewordbreak\discretionary}{(-){}\prewordbreak
      \fi
    \else\ifx)\next
      \ifx\nextnext\blankspace
        \prewordbreak)\relax
      \else
        \prewordbreak\discretionary{-)}{)}\postwordbreak
      \fi
    \else\ifx'\next
      \prewordbreak\discretionary{-}{'}\postwordbreak
    \else\ifx<\next
      \hbox{\beginofsubsentence}\prewordbreak
    \else\ifx>\next
      \prewordbreak\endofsubsentence
    \else
      \checkafterdiscretionary
      \prewordbreak\hbox{\betweendiscretionary\nextnext}\postwordbreak
    \fi\fi\fi\fi\fi\fi\fi\fi
  \nextnextnext}

9 \def\checkafterdiscretionary%
  {\ifx,\nextnext
  \def\nextnextnext{\afterassignment\egroup\let\next=}%
  \else
  \let\nextnext=\relax
  \fi}
```

Before we show some more tricky alternative, we first install the mechanism:

```
10 \installdiscretionaries || \compoundhyphen
```

One of the drawbacks of this mechanism is that characters can be made active afterwards. The next alternative can be used in such situations. This time we don't compare the arguments directly but use the `\uccode`'s instead. `TEX` initializes these codes of the alphabetic glyphs to their uppercase counterparts. Normally the other characters remain zero. If so, we can use the `\uccode` as a signal.

The more advanced mechanism is activated by calling:

```
\enableactivediscretionaries
```

which is defined as:

```
11 \def\enableactivediscretionaries%
    {\uccode'=(\relax \uccode')=\relax \uccode'=='\relax
     \uccode'<=<\relax \uccode'>=>\relax
     \uccode'=''\relax \uccode'~='~\relax
     \let\dottextmodediscretionary = \activedottextmodediscretionary
     \let\dododottextmodediscretionary = \activedododottextmodediscretionary}
```

We only have to redefine two macros. While saving the \uccode in a macro we have to take care of empty arguments, like in ||.

```
12 \def\activedottextmodediscretionary#1%
    {\bgroup
     \def\dodottextmodediscretionary##1#1%
       {\def\betweendiscretionary{##1}%
        \def\nextuccode####1####2\relax%
         {\ifcat\noexpand####1\noexpand\relax
          \edef\nextuccode{0}%
          \else
           \edef\nextuccode{\the\uccode'####1}%
          \fi}%
         \nextuccode##1@\relax
         \futurelet\nextnext\dododottextmodediscretionary}%
     \let\discretionarycommand=#1%
     \def\textmodediscretionary{\getvalue{textmodediscretionary}\string#1}}%
     \futurelet\next\dodottextmodediscretionary}
```

This time we use \ifnum:

```
13 \def\activedododottextmodediscretionary%
    {\let\nextnextnext=\egroup
     \ifx\discretionarycommand\next
       \checkafterdiscretionary
       \prewordbreak\hbox{\textmodediscretionary\nextnext}\postwordbreak
     \else\ifnum\uccode'==\nextuccode
       \prewordbreak\textmodediscretionary
     \else\ifnum\uccode'~=\nextuccode
       \prewordbreak\discretionary{-}{\thinspace}\postwordbreak
     \else\ifnum\uccode'(\=\nextuccode
       \ifdim\lastskip>!!zeropoint\relax
         (\prewordbreak
         \else
          \prewordbreak\discretionary}{(-){}\prewordbreak
         \fi
       \else\ifnum\uccode')=\nextuccode
         \ifx\nextnext\blankspace
           \prewordbreak)\relax
         \else
          \prewordbreak\discretionary{-}{\}{})\postwordbreak
         \fi
       \else\ifnum\uccode'='\nextuccode
         \prewordbreak\discretionary{-}{\}'\postwordbreak
       \else\ifnum\uccode'<=\nextuccode
         \hbox{\beginofsubsentence}\prewordbreak
       \else\ifnum\uccode'>=\nextuccode
         \prewordbreak\endofsubsentence
       \else
         \checkafterdiscretionary
```

```

\prewordbreak\hbox{\betweendiscretionary\nextnext}\postwordbreak
\fi\fi\fi\fi\fi\fi\fi\fi
\nextnextnext}

```

Now we can safely do things like:

```

\catcode'<=\@active \def<{hello there}
\catcode'>=\@active \def>{hello there}
\catcode'(\@active \def({hello there}
\catcode')=\@active \def){hello there}

```

In normal day-to-day production of texts this kind of activation is seldom used.² If so, we have to take care of the math mode explicitly, just like we did when making | active. It can be confusing too, especially when we load macropackages afterwards that make use of < in \ifnum or \ifdim statements.

14 \protect

²In the ConTeXt manual the < and > are made active and used for some cross-reference trickery.

Een moderne toren van Babel

Johannes L. Braams

Samenvatting

In deze voordracht zal ik kort aandacht besteden aan ‘wat is \TeX ’, ‘wat is \LaTeX ’. Hierbij komt ook het concept van ‘structuur markup’ aan de orde. Vervolgens zal ik ingaan op de geschiedenis van het **babel** pakket. Aan de hand van een aantal voorbeelden zal ik toelichten welke problematiek zich voordoet bij het op een correcte wijze vormgeven van documenten waarin meerdere talen worden gebruikt.

Een moderne toren van Babel Overzicht

Overzicht

1. Inleiding
2. Wat is \TeX en \LaTeX
3. Wat is structuur markup?
4. De geschiedenis van babel
5. Meertalige documenten

21 November 1996 \TeX niek 2

1 Inleiding

Wanneer een auteur een document moet maken staan hem daartoe in principe twee verschillende *typen* hulpmiddelen ter beschikking.

Talen en Tools Inleiding

Inleiding

Tekstproductie:

WYSIWIG of
markup?

21 November 1996 \TeX niek 3

Het meest gebruikte type hulpmiddel is de WYSIWIG¹ tekstverwerker. Voorbeelden hiervan zijn ‘Word for Windows’ en Wordperfect for Windows’. Deze hulpmiddelen vereisen een grafische user interface en zijn derhalve voornamelijk voor personal computers² beschikbaar.

In de Unix wereld zijn hulpmiddelen van het tweede type, de ‘markup taal’ beter beschikbaar. Voorbeelden hiervan zijn troff en \TeX / \LaTeX .

Tussen de gebruikers van beide typen ‘tekstverwerkers’ woedt soms net zo’n heftige ‘geloofs’-strijd als tussen de gebruikers van VMS en Unix. Een voordeel van het eerste type hulpmiddel is dat de gebruiker directe terugkoppeling krijgt over hoe zijn document er op papier ongeveer uit zal gaan zien. Een voordeel van het tweede hulpmiddel is dat de auteur zich op enig moment (tot op zekere hoogte) alleen met de inhoud (en niet, of in ieder geval minder, met de vorm) van zijn document bezig houdt. De vorm is dus voor een belangrijk deel losgekoppeld van de inhoud.

2 Wat is \TeX en \LaTeX ?

\TeX is een programma voor het ‘zetten’ van teksten, ontwikkeld door professor Donald E. Knuth van de Stanford Universiteit. De auteur heeft \TeX ontwikkeld omdat de drukkerij die zijn serie boeken ‘The Art of Computer Programming’ drukte met andere apparatuur ging werken waardoor zijn boeken een ander aanzien kregen. Hij heeft het als een uitdaging gezien om met behulp van de computer minstens dezelfde typografische kwaliteit te bereiken die voorheen met door een zetter de zetmachine werd geleverd. Omdat ook het lettertype waarin zijn boeken gezet werden niet meer beschikbaar was heeft hij ook het programma METAFONT ontwikkeld waarmee hij het lettertype ‘Monotype Modern 8a’ als ‘computer modern roman’ opnieuw het licht deed zien.

¹WYSIWIG staat voor ‘What you see is what you get’. In de praktijk blijkt dit vaak net niet helemaal op te gaan. . .

²in de breedste betekenis; een Apple Macintosh valt voor mij ook onder de ‘personal computers’.

Wat is T_EX?

T_EX is een programma om
met behulp van de computer
hoge kwaliteit zetwerk voor
The Art of Computer Programming
te produceren.

Professor Knuth ontwikkelde een programma, T_EX (spreek uit: ‘tech’) genaamd, dat als invoer de tekst van een document krijgt, vermengd met instructies die het programma duidelijk maken hoe de tekst op papier moet worden gezet. Teneinde het gebruik van T_EX zo flexibel mogelijk te laten zijn heeft de auteur T_EX voorzien van een macro taal. Dat wil zeggen: het programma T_EX zelf kent een beperkt aantal instructies (ongeveer 300 zogenaamde primitives). De instructieset is eenvoudig uit te breiden door nieuwe instructies te definiëren op basis van bestaande instructies. Professor Knuth heeft met het beschikbaar stellen van T_EX ook een wat uitgebreidere instructieset (ook wel macropakket genoemd) ter beschikking gesteld, ook wel bekend onder de naam PLAIN T_EX. Later zijn hierop door verschillende andere auteurs uitbreidingen op gemaakt. De bekendste, en met name in Europa, meest verspreide uitbreiding is het door Leslie Lamport ontwikkelde L^AT_EX.

Wat is L^AT_EX?

L^AT_EX is een uitbreiding op
de instructieset van T_EX.

L^AT_EX kent een taal die
de *structuur* van
een document beschrijft.

Eén van de belangrijkste kenmerken van L^AT_EX is dat de ‘taal’ van L^AT_EX op een ander niveau is gedefinieerd. Vergeleken met programmeertalen is de verhouding tussen PLAIN T_EX en L^AT_EX overeenkomstig die tussen assembler en PASCAL.

3 Wat is structuur markup?

De term ‘markup’ wordt gebruikt voor commando’s die tussen de normale tekst worden geplaatst en die als instructie gelden voor diegene die ervoor moet zorgen dat de tekst vermenigvuldigd kan gaan worden. In het Nederlands wordt ook wel de term ‘zetinstructies’ gebruikt,

hoewel die term naar mijn idee voornamelijk voor de één na laatste stap in het productieproces van toepassing is. Zetinstructies schrijven bijvoorbeeld voor dat een hoofdstuk in een boek op een rechter pagina moet beginnen; dat de hoofdstuktitel een bepaalde afstand naar beneden op de pagina gezet moet worden uit een specifiek corps, etc. Dergelijke markup wordt ook wel visuele markup genoemd.

Wat is structuur markup?

- Visuele markup


```
‘start een rechter pagina’;
‘skip 25 mm witruimte’;
‘zet het hoofdstuknummer uit Arial corps
18’
2
‘skip 15 mm witruimte’;
‘zet de hoofdstuktitel uit Arial corps
20’
De titel van het hoofdstuk
‘skip 15 mm witruimte’
```

Wanneer de commando’s die tussen de tekst zijn gevoegd niet meer direct de te bereiken visuele effecten op de pagina beschrijven, maar in hoofdzaak iets zeggen over de ‘logische elementen’ van een document spreken we over ‘structuur markup’. Een dergelijke taal bevat onder meer commando’s waarmee aangegeven kan worden dat een nieuw hoofdstuk begint en wat de titel is van dat hoofdstuk, waar een genummerde lijst begint en eindigt, waar de afzonderlijke elementen van die lijst beginnen etc. Een document dat op een dergelijke wijze van markup is voorzien is dan ook veel beter geschikt om hergebruikt te worden in bijvoorbeeld database toepassingen. Doorgaans wordt onafhankelijk van het document dat de markup bevat, bepaald hoe de gebruikte commando’s leiden tot bepaalde visuele effecten op papier (of op een beeldscherm).

Wat is structuur markup?

- Structuur markup


```
‘doctype:artikel’
‘taal:nederlands’
‘chapter:De titel van het hoofdstuk’
```

Bovenstaand voorbeelden illustreren naar mijn idee een aantal belangrijke verschillen tussen visuele markup en structuur markup. In het tweede geval is het 100% zeker dat *elke* hoofdstuktitel er consequent hetzelfde uitziet. In het eerste geval kan makkelijk een fout gemaakt worden wat uiteindelijk een slordige indruk op de lezer zal maken.

In L^AT_EX zijn dergelijke ideeën over ‘structuur markup’ in belangrijke mate terug te vinden.

4 De geschiedenis van babel

Tegen de tijd dat de populariteit van het programma T_EX zodanig groot was dat ook mensen in Europa ermee aan de slag gingen had Leslie Lamport L^AT_EX al uitgevonden. Het gevolg was dat in Europa relatief meer gebruik gemaakt werd (en wordt) van L^AT_EX dan van PLAIN T_EX. Maar L^AT_EX is een van oorsprong Amerikaans produkt. Dus een hoofdstuk begon altijd, hard gecodeerd, met het woord ‘Chapter’; *ook* als het om een nederlandstalig document ging. Hiervoor werden in Europa op verschillende plaatsen door verschillende mensen gelijksoortige oplossingen voor bedacht. Met name in Duitsland en Frankrijk waren aanpassingen van L^AT_EX in omloop die aan dit soort problemen het hoofd trachtten te bieden. In die tijd was ook in Nederland een schare enthousiaste gebruikers van L^AT_EX ontstaan die zich opnieuw met dit probleem geconfronteerd zagen. De eerste ‘oplossing’ was om de uitbreidingen die voor het Duits in Duitsland al waren gerealiseerd voor het Nederlands over te nemen (voor zover relevant). Hierin ligt de oorsprong van het babel pakket.

Talen en Tools Het ontstaan van Babel

Het ontstaan van Babel

- Verschillende, gelijksoortige, *nationale* uitbreidingen voor de ondersteuning van de eigen taal door L^AT_EX;
- Veel gezamenlijke programmatuur;
- *Babel*: Breng alle gezamenlijke programmatuur onder in één gemeenschappelijke ‘kernel’ en laat de ondersteuning voor de verschillende talen daar gebruik van maken.

21 November 1996 T_EXniek 8

Op de EuroT_EX conferentie in Karlsruhe (1989) begonnen mijn ideeën meer vorm te krijgen, met name naar aanleiding van discussies met een aantal conferentie deelnemers. In de eerste helft van 1990 kwam de eerste release van *babel* beschikbaar. Op dat moment was er ondersteuning voor een half dozijn talen. Niet lang na het verschijnen van *babel* begonnen mensen bijdragen te leveren om het aantal ondersteunde talen te laten groeien. Momenteel worden 31 verschillende talen ondersteund.

Talen en Tools Ontwikkeling van Babel

Ontwikkeling van Babel

- Eerste versie in 1990; 6 talen worden ondersteund.
- Tweede versie in 1992; meer talen, ‘stack’ van macro definities ten behoeve van ‘genest’ talen.
- Derde grote aanpassing 1994–1995; nog meer talen; introductie van het begrip ‘shorthand’.

21 November 1996 T_EXniek 9

De ondersteuning voor sommige van die talen gaat niet verder dan het ter beschikking stellen van de juiste vertalingen voor woorden als ‘chapter’, ‘figure’ etc. Voor andere talen gaat de ondersteuning veel verder; soms wordt de invoer van letters met diacritische tekens vereenvoudigd, soms worden speciale typografische gewoonten ondersteund.

Talen en Tools Hoofdfuncties van Babel

Hoofdfuncties van Babel

- Kiezen van de juiste afbreekregels;
- Kiezen van de juiste vertalingen van standaardwoorden;
- Kiezen van de juiste vertaling en vorm van de datum;
- Inschakelen van taal-specifieke conventies;

21 November 1996 T_EXniek 10

Talen en Tools taal-specifieke conventies

taal-specifieke conventies

- Vereenvoudiging van invoer,
- Speciale afbreekregels (zoals verdubbeling van letters of verdwijnen van een trema bij afbreken),
- bijzondere effecten.

21 November 1996 T_EXniek 11

De belangrijkste functie van *babel* is het selecteren van het juiste afbreek algoritme. Zoals in figuur 1 te geïllustreerd wordt is dat tamelijk belangrijk. (Kijk er de dagbladen maar op na!) In dezelfde figuur wordt ook de tweede functie van *babel* getoond, namelijk de vertaling van het woord ‘figure’ in het rechter deel van de figuur. Eén van de taal-specifieke conventies voor het Nederlands tenslotte, is de ondersteuning van het lage aanhalingsteken openen (,,).

5 Meertalige documenten

Er kunnen twee klassen van meertalige documenten onderscheiden worden. Ten eerste zijn er documenten waarin een ‘hoofdtal’ wordt gebruikt, maar waarin ook passages uit één of meer andere talen voorkomen. Voor dergeli-

Voorbeeld

De
geïrri-
teerde
mede-
wer-
ker
ge-
bruikte
een
tek-
stver-
werker.

Figure 1: Nederlandse tekst, engels afgebroken

De
geïrri-
teerde
mede-
wer-
ker
ge-
bruikte
een
tekst-
ver-
wer-
ker.

Figuur 2: Nederlandse tekst, nederlands afgebroken

Figuur 1: Vergelijking tussen het afbreken van nederlandse tekst met een engels en een nederlands afbreek algoritme

jke documenten zijn de vertalingen van de standaardwoorden minder van belang, maar de auteur kan er aan hechten wél zoveel mogelijk de juiste typografische conventies te hanteren voor elke taal. Tenminste moet gezorgd worden dat woorden zoveel mogelijk correct worden afgebroken.

Het programmaboekje van deze conferentie kan als voorbeeld dienen van een dergelijk document. Het bevat enkele abstracts in het Engels, terwijl de het grootste deel van de tekst in het Nederlands is gesteld.

Een tweede klasse van documenten is die waarin twee of meer talen een min of meer gelijkwaardige rol spelen. Voorbeelden van dergelijke documenten kunnen met name gevonden worden bij de EU. Sommige publicaties bevatten dezelfde inhoud meermaals, in verschillende talen. Andere gebruiken voor elk hoofdstuk een andere officiële taal.

Meertalige documenten

1. Eén 'hoofdtaal' met passages uit andere talen
2. Meerdere gelijkwaardige 'hoofdtalen'

Stretching the Limits of Babel, an Ever Growing Package?

Johannes L. Braams
 \TeX niek

1 Introduction

This presentation starts with an overview of the history of `babel`. The current state of `babel` is described and some thoughts on the problem areas that need to be dealt with are presented. In the last part of the presentation I will discuss possible future directions of development.

Stretching the Limits of babel

Overview

1. Introduction
2. A brief history of babel
3. The current state of babel
4. (Un)solved problems
5. Future development

March 1997

JLB

2

2 A brief history of babel

The first ideas for developing a set of macros to support typesetting documents with \TeX in languages other than English developed around the time of the Euro \TeX conference in Karlsruhe (1989). Back then I had created support for typesetting in Dutch by stealing `german.tex` (by Hubert Partl c.s.) and modifying it for Dutch conventions. This worked, but I was not completely satisfied as I hate duplication of code. Soon after that I found that more ‘copies’ of `german.tex` existed to support other languages. This led me to the idea of creating a package that combines these kind of language support packages. It would have to consist of at least two ‘layers’: all the code the various copies of `german.tex` had in common in one place, loaded only once by \TeX , and a set of files with the code needed to support language specific needs. During the Karlsruhe conference the name ‘`babel`’ came up in discussions I had. It seemed an appropriate name and I stucked to it.

Stretching the Limits of babel

A brief history of babel

A brief history of babel

- First ideas at Euro \TeX ’89 Karlsruhe
- First published in TUGboat 12–2
- Update article in TUGboat 14–1
- Presentation at Euro \TeX ’95

March 1997

JLB

3

After the conference I started to work on “`babel`, a multi-lingual style-option system for use with \LaTeX ’s standard document styles”. The first release with support for about half a dozen languages appeared in the first half of 1990. In TUGboat volume 12 number 2 an article appeared describing `babel`. Soon thereafter people started contributing translations for the ‘standard terms’ for languages not yet present in `babel`. The next big update appeared in 1992, accompanied by an article in TUGboat volume 14 number 1. The main new features were that an interface was added to ‘push’ and ‘pop’ macro definitions and values of registers. Also some code was moved from language files to the core of `babel`. In 1994 some changes were needed to get `babel` to work with $\LaTeX_{2\epsilon}$. As it turned out a lot of problems were still unsolved, amongst which the incompatibility between `babel` and the use of T1 encoded fonts was most important.

In 1995 the concept of ‘shorthands’ was introduced. A ‘shorthand’ is a construction based on active characters, but the active character changes its definition according to its context. It can have an argument and it can get a definition on user level, language level or system level.

In 1996 support for languages that need fonts with a different encoding was (re)introduced. Before the advent of $\LaTeX_{2\epsilon}$ `babel` contained support for typesetting Russian texts, using cyrillic fonts. This support had to be completely rewritten.

3 The current state of babel

3.1 Languages supported

Currently `babel` supports no less than 36 languages. The level of support for the various languages varies. For some languages the support is nothing more than a provision of the translation of (most) words that can be generated by \LaTeX . For other languages shorthands are defined to ease

the typing of texts or to support certain hyphenation tricks. For some languages a fontencoding switch or specific typographic conventions need to be supported. The support for the Greek language also provides a different enumerating scheme (`\greeknumerals`).

For a number of languages multiple variants are supported.

The languages directly supported by `babel` are shown in the following slide.

Stretching the Limits of babel		The languages supported by babel	
The languages supported by babel			
Afrikaans	English	Irish	Sanskrit
Bahasa	Esperanto	Italian	Scottish
Basque	Estonian	Kannada	Spanish
Breton	Finnish	Lower Sorbian	Slovakian
Catalan	French	Norwegian	Slovene
Czech	Galician	Polish	Swedish
Croatian	German	Portuguese	Turkish
Danish	Greek	Rumanian	Upper Sorbian
Dutch	Hungarian	Russian	Welsh

March 1997 JLB 4

Apart from these languages two separate distributions are known to exist that are based on `babel` and provide support for the Ethiopian and Ukrainian languages.

3.2 Language attributes supported

At the EuroTeX 95 conference a `babel` BOF was held. The discussion focused on the topic of what defines a language. In the end a list of language attributes was produced.

Stretching the Limits of babel		Language attributes	
Language attributes			
1. Hyphenation patterns and associated <code>\lefthyphenmin</code> and <code>\righthyphenmin</code> .			
2. Captions and dates			
3. Quotation marks			
4. Typographic conventions			

March 1997 JLB 5

For attribute 2 it was thought that perhaps several formats of dates might need to be supported for some languages.

Stretching the Limits of babel		Language attributes (ctd)	
3			
1. Fontencoding			
2. Mathematics			
3. Enumerating			
4. Ligatures			
5. Punctuation			

March 1997 JLB 6

The font encoding is also referred to as output encoding. An example of the language (or nationality) dependency of mathematics (2) is that `\tan` needs to produce either *tan* or *tg*.

Stretching the Limits of babel Language attributes (ctd)

- 4
1. Input encoding
2. Direction of writing
3. Hyphen split
4. Collating sequence (`\alph` etc.)
5. Conventions for emphasis

March 1997 JLB 7

Jiří Zlatuška has published an article about ‘hyphen split’ which I couldn’t trace. The conventions for emphasis might possibly be better placed in a document class which implements publishing house conventions.

For a large number of the attributes above, examples can be found in `babel`. The attributes 1 and 2 were the very basis of the system and are supported for all the languages in the `babel` distribution. For a large number of languages some support is available for non-standard quotation marks (3) and specific typographic conventions (4).

The attributes 1 through 4 are less common, but do occur for some languages. The attributes 4 through 5 do not currently occur in any language definition file. People have been trying to get support for Hebrew typesetting working over the past couple of years. For this they need attribute 2 to be supported. The work that I am aware of so far has shown that bidirectional typesetting needs extensive changes in L^AT_EX itself which can not easily be done from the outside of L^AT_EX.

3.3 Document elements supported

In a document various elements can be identified which should possibly inherit the language attribute.

Stretching the Limits of babel		Document elements	
Document elements			

- the main text
- table of contents (and of figures, tables)
- running headers (and footers)
- floating objects (`marginpars`, figures, tables)

March 1997 JLB 8

In the original `babel` system only the main body of the text would be influenced by the setting of a language switch. During the recent history of `babel` the tables of contents etc. and the running headers have been added. They now inherit the language attribute which is valid at the time an

entry in the table of contents is generated. In the process of adding this support the setting of the language attribute has also been added to L^AT_EX's auxiliary files.

3.4 Shorthands

From the start of `babel` some language definition files have contained code to make some characters 'active'. In the early years this only happened to the double quote ("), which was a rather safe choice as both Don Knuth and Leslie Lamport had stated in their books that it should not be used in texts. Nevertheless this active character caused problems as it also has a function to indicate to T_EX that a hexadecimal number follows.

With `babel` release 3.5 in 1995 the concept of 'shorthands' was introduced. A shorthand is basically an active character, possibly followed by a second character.

Stretching the Limits of babel shorthands

shorthands

- A shorthand consists of an active character, possibly followed by an argument
- Shorthand characters do not change `\catcode`
- Shorthand characters are written out unexpanded

March 1997 JLB 9

The difference with earlier releases of `babel` was that from then on active characters remain active throughout the document. They do not change `\catcode` other than in controlled situations, such as a `verbatim` environment. The only thing that changes is their definition.

When necessary, shorthand characters are made to expand to a non-active copy of themselves. Another aspect of shorthands is that when they are written out (to an `.aux` file for instance) they do not get expanded.

Currently quite a large number of characters are used as shorthand characters, as can be seen in the following slide.

Stretching the Limits of babel Overview of shorthands

- ~ System, Basque, Catalan, Estonian, Galician, Sanskrit, Spanish
- : Breton, French, Russian, Turkish
- ; Breton, French, Russian
- ! Breton, French, Russian, Turkish
- ? Breton, French, Russian
- " Basque, Catalan, Danish, Dutch, Estonian, Finnish, Galician, German, Polish, Portuguese, Russian, Sanskrit, Slovene, Spanish, Swedish, Upper Sorbian
- ‘ Catalan (optional)
- ’ Catalan, Galician, Spanish (optional)
- ^ Esperanto
- = Turkish
- _ Sanskrit

March 1997 JLB 10

4 Unsolved problems

Stretching the Limits of babel Unsolved problems

Unsolved problems

1. floats inheriting language attributes
2. bidirectional typesetting
3. Multiple input encodings and hyphenation
4. Non standard input encoding

March 1997 JLB 11

In the current `babel` a number of problems remain unsolved. The most important ones are shown in the slide above.

Problem 1 still needs to be researched. It seems obvious that a floating object should inherit the language attribute from the 'surrounding text'. This probably means that a way has to be found to pass this information to the floating object.

On problem 2 Rama Porrat in Israel has done quite a lot of work. As far as I know she didn't quite succeed in creating a `hebrew.sty` though. She did find out however that L^AT_EX needs to be changed in a number of points to fully support right-to-left typesetting. L^AT_EX is by nature a package which was developed in an environment where virtually everybody uses solely left-to-right typesetting.

With multilingual documents the authors will no doubt find out that the hyphenation is done by T_EX on a paragraph basis. This means that the hyphenation algorithm uses the `\lccodes` which are in effect at the end of the paragraph for the hyphenation process of the entire paragraph. In some cases this might lead to wrong hyphenation of single words or phrases in a different language for which other `\lccodes` are needed.

With $\text{\LaTeX}2_{\epsilon}$ the (still experimental) package `inputenc` came along. This package makes all the ‘special’ characters active and defines them to expand to \LaTeX ’s ‘internal’ encoding. For some languages people have claimed that for them the ‘special’ characters need to be of the category code ‘letter’.

5 Future development

Stretching the Limits of babel

Possible future development

Possible future development

- Extending the support with more languages?
- Integrating the Ω version of babel with the ‘normal’ babel distribution
- Integrating concepts of babel in \LaTeX ?

As people keep sending me contributions for ‘new’ languages the number of languages supported by `babel` will keep growing. Perhaps the support for some languages will be distributed separately from the core `babel` distribution (which would ease the task of maintenance of `babel` somewhat).

Currently a special version of `babel` has been developed by Yannis Haralambous for multilingual support with Ω . Both versions of `babel` should be reunited for the sake of easier maintenance.

The experience gained with the development and maintenance of `babel` over the past years is very valuable in the development of the language support module for $\text{\LaTeX}3$. Some of the *concepts* that have been developed for `babel` will form input for the development of this part of $\text{\LaTeX}3$.