

Verlag Verlag 19^e ntg-bijeenkomst, 12 juni 1997

Nederlandstalige
T_EX
Gebruikersgroep

aanwezig

A. Al-Dhahir (UT); A. Bakker H. Bakker (UvA); A.W.W.M. Biegstraaten (TUD); T. Bloo (De Jonge Onderzoekers); J.L. Braams (T_EXniek); W.T. Bril (TNO-WT); L. de Coninck (de Kraal); W.H. Dekker (Servalys); G. van den Dobbelaer (LOGIN BV); W. Dol; W. Dolman (Hogeschool van Amsterdam); P. Eilers (DCMR); G. Eryvynck (Kath. Universiteit Leuven); E. van Eynde (K.U. Leuven); C.M. Fortuin (Hogeschool Gelderland); E.H.M. Frambach (RUG); G. Geens (Universiteit Gent); M. Gelderman; W.L. Gilberts (Hogeschool van Amsterdam); F. Goddijn (FGBBS); W.J. van de Guchte; J. Hagen (Pragma); J. Hellings (UvA-AMC); R. van der Heijden (Hogeschool van Utrecht); T.C.A.J. Hoekwater (Kluwer); P. de Jong (TUD); N.S. Kroonenberg (RUG); R. Kwee C.G. van der Laan P.G. Miedema (Koninklijke Militaire Academie); C.C.M. Moes (TUD); G.J.H. van Nes (ECN); G. Oomen (Kluwer); P. van Oostrum (UU); A.F. Otten; S.A.M. Pepping (Elsevier); J.W.M. Pijnenburg (KUB); R. Rietman J. Rusch (Philips Nat. Lab.); A. van Ryckeghem (KHBO); R. van Rijn (Philips Nat. Lab.); A.C.A. Serier W. Smit A.B. Soos (UT); J.H. van de Stadt E.H.M. Ulijn (TUD); H. Vink J.E. van Weerden (UU); J.J. Winnink P. van der Wijden (TUD);

notulist

Frans Goddijn

1 Opening

Voorzitter Erik Frambach opent de achttiende NTG-bijeenkomst en dankt gastheren Ton Biegstraaten en Peter van der Wijden.

2 Notulen van de achttiende ntg-bijeenkomst van 24 oktober 1996

De notulen worden ongewijzigd vastgesteld.

3 Ingekomen stukken en mededelingen

Secretaris Gerard van Nes verwijst naar de leestafel annex NTG-winkel die in de hal is ingericht. Daar zijn de binnengekomen tijdschriften van diverse zusterverenigingen te vinden en een gevarieerd aanbod van voorbeelden van

boeken en documenten die met TeX zijn gemaakt. Ook Addison-Wesley werd bereid gevonden een stand in te richten waar een veelheid aan TeX-boeken en andere boeken die voor ons van belang kunnen zijn, met korting worden aangeboden. Het zojuist verschenen boek "The LaTeX Graphics Companion" ligt er in voldoende exemplaren gereed. Gerard heeft gezorgd voor de gekleurde naambadges: blauw voor gastheren en sprekers, geel voor ereleden, rood voor bestuur en wit voor overige leden, waarbij de aansporing hoort dat leden met witte badges een volgend keer een andere kleur badge kunnen "verdienen". . .

4 NTG Ledenvergadering

4.1 Jaarverslag secretaris

Piet van Oostrum vraagt of 4allTeX nu een toekomst bij Addison Wesley gaat krijgen of niet. Erik Frambach antwoordt dat dit nog niet zeker is maar dat de verkoop intussen goed doorloopt.

4.2 Jaarverslag penningmeester

Hierbij wordt toegelicht dat de winst van het 4allTeX project nog niet is gespecificeerd omdat het precieze bedrag nog moet worden vastgesteld. Over de verdeelsleutel (50-50) zijn 4TeX-team en bestuur het al eerder eens geworden.

4.3 Royementen

De vergadering gaat hiermee akkoord.

4.4 Verslag kascontrolecommissie

Kees van der Laan meldt namens de commissie dat men akkoord is, onder voorbehoud van een nog uit te voeren steekproefsgewijze controle van een aantal "bonnetjes". De commissie adviseert overigens een eenvoudiger manier van administreren. De penningmeester licht toe dat hij hiermee al ver is gevorderd.

4.5 Vaststelling nieuwe kascontrolecommissie

Kees van der Laan en J.H. van de Stadt zullen samen de nieuwe commissie vormen.

4.6 Voorstel erelidmaatschap

Een ingevoegd agendapunt: het bestuur stelt voor om Piet van Oostrum te benoemen tot erelid voor het leven. Hij komt daarmee in gezelschap van Kees van der Laan en Jo-

hannes Braams. De reden om Piet erelid te maken is zijn niet aflatende inzet bij het helpen van medegebruikers van TeX. Onder andere via mailinglijst TEX-NL kunnen talloze gebruikers van TeX gebruik maken van zijn grote expertise en zijn duidelijke, snelle, direct toepasbare uitleg. Dankzij zijn aanwijzingen heeft menig een de weg gevonden naar de oplossing of het macropakket dat het gewenste resultaat gaf. Zijn eindeloze geduld en snelle support zijn van onschatbare waarde voor de beginnende én de ervaren TeX-gebruiker.

De vergadering gaat bij acclamatie akkoord met dit voorstel en Piet van Oostrum krijgt een nieuwe badge in bijpassende kleur uitgereikt. Bij dit erelidmaatschap hoort het voorrecht om levenslang kostenloos lid te mogen blijven van de NTG.

4.7 Bestuursverkiezingen

Er wordt afscheid genomen van Wietse Dol als bestuurslid en penningmeester. Een cadeau wordt Wietse namens alle leden aangeboden.

De vergadering gaat akkoord met de benoeming van Wybo Dekker als opvolger. Wybo Dekker licht toe dat hij in de afgelopen maanden zich onder andere heeft beziggehouden met het omzetten van de tot dan toe gebruikte Lotus Spreadsheet naar een simpeler database die door een PERL script op meerdere manieren wordt verwerkt tot de benodigde TeX-documenten.

4.8 Verslag hoofdredacteur MAPS

Erik Frambach dankt allereerst Taco Hoekwater voor de energieke wijze waarop hij zijn taak als hoofdredacteur heeft aangepakt. Dank komt ook toe aan Taco's werkgever Kluwer, die hem in de gelegenheid stelt om mede onder werktijd met de MAPS zaken bezig te zijn. Taco Hoekwater licht toe dat het inderdaad erg veel werk blijkt te zijn, te veel voor de huidige redactie, en dat aanvulling dringend is gewenst.

Opbouwende kritiek kan het beste per post of per email worden gemeld. Het zou wenselijk zijn om auteurs tevoren nog een preview te geven van hun eigen artikel, maar in de praktijk ontbreekt daarvoor veelal de tijd. De MAPS.STY zal grondig worden gereviseerd om het mogelijk te maken dat meer artikelen met minder problemen kunnen worden verwerkt.

Siep Kroonenberg meldt zich aan als nieuw lid van de redactie.

4.9 Wat verder ter tafel komt

MAPS Award Gerard van Nes meldt dat er veel leuke, afwisselende artikelen zijn verschenen in deze MAPS. Een aantal van de auteurs heeft al eens een MAPS-award ge-

kregen. Gerard van Nes wijst nog eens op de verdiensten van erelid en oprichter Kees van der Laan, met 5 artikelen in deze MAPS aanwezig, en Hans Hagen zelfs met 6 bijdragen.

De MAPS Award gaat ditmaal naar Ton Biegstraaten voor zijn spectaculaire artikel over fonts, een uitdagend, stimulerend en (voor de redactie) technisch zeer gecompliceerd artikel dat deze MAPS met name tot een bijzondere MAPS heeft gemaakt. De *stretchable sweet boxes* van gesuikerde glue zijn er tevens voor Gerrit Oomen, van Kluwer, die aan Taco Hoekwater de ruimte en stimulans heeft gegeven om zich zo voor de MAPS in te spannen.

Vestzakboekje Kees van der Laan stelt voor een vestzakboekje te produceren met daarin een versie van bovengenoemd artikel over fonts. Taco Hoekwater voelt daar wel voor en zal in dat geval een artikel toevoegen over de diverse problemen die deze op CTAN beschikbare fonts opleveren.

Verouderde CM fonts Hans Hagen vraagt zich af of de MAPS redactie nog rekening moet houden met verouderde CM fonts. Op Knuths homepage wordt daarover nogal gemopperd. Taco Hoekwater licht toe dat dit probleem al goeddeels is "overgewaaid".

4.10 Miscellaneous

Communicatiezaken: TEX-NL, WWW, FTP-server, FGBBS Jules van Weerden merkt op dat de 4TEX lijst de laatste tijd erg rustig is. Er is een afspraak met het rekencentrum dat er wel meerdere mailinglijsten kunnen worden opgezet zolang deze niet te omvangrijk worden.

Piet van Oostrum vertelt dat het wachten is op een nieuwe, betere FTP-server. Hij licht verder de technische verschillen (gemak, transmissiesnelheid) toe tussen benadering via FTP en via HTTP.

Frans Goddijn merkt op dat er in de MAPS nog een oud nummer voor FGBBS wordt vermeld: dit is sinds de overgang naar tiencijferige nummers **026-3217041** geworden. Verder zoekt hij assistentie van iemand die met enige kennis van zaken een aantal files en directories van CTAN kan selecteren en van commentaar voorzien ten behoeve van het plaatsen van deze files op FGBBS. Jos Winnink belooft hiervoor zijn best te zullen doen.

TUG/LUG aangelegenheden Erik Frambach en Hans Hagen zullen naar de TUG-bijeenkomst gaan deze zomer. Kees van der Laan meldt dat er in september een Cyrtug bijeenkomst is te verwachten. Informatie hierover kan bij Kees van der Laan worden ingewonnen. De MAPS-redactie verwacht een dik verslag over deze bijeenkomst!

De nieuwe UNIX TeX CD-ROM Wybo Dekker merkt op dat er een probleem is met DVIPS van deze CD-ROM, dat buggy werkt indien er gebruik wordt gemaakt van *partial font downloading*. Taco Hoekwater meldt dat de win32 versie wel compileert, maar dat er geen viewer bijzit, waarop Erik Frambach uitlegt dat hier een reden voor is: previewen is een *externe* gebeurtenis en de programmatuur ervoor hoort ook niet bij de standaard programmatuur. Pieter Miedema meldt dat de Linux versie wel van CD-ROM is te draaien, maar zich niet van HD laat gebruiken.

Vooruitzichten nieuwe 4allTeX CD Wietse Dol licht toe: een manco van emTeX is dat het zeer traag werkt met de nieuwe TDS (TeX directory structuur). Het pakket is daarentegen wel betrouwbaar en stabiel. Het plan is om 4allTeX te updaten en uit te breiden met nieuwe opties (alternatieve viewers) en dit *op korte termijn* te realiseren. Op langere termijn wordt de TDS geïmplementeerd en dan zal de CD-ROM een geheel andere opzet krijgen. In 1998 valt daarna de geheel nieuwe Windows-versie te verwachten. In de pauze zal Wietse de nieuwigheden van de komende 4allTeX CD-ROM alvast laten zien. Het 4allTeX-team houdt zich aanbevolen voor tips en hints, o.a. betreffende de vindplaatsen van nu nog onbekende tools (immers, niet alle goeie TeX-spullen staan ook op CTAN. Prachtige dingen blijken soms heel ergens anders vandaan te komen).

Erik Frambach vertelt dat de contacten met Addison-Wesley over het 4allTeX project lang niet zo vlot gaan als werd gehoopt. De komende update van de CD-ROM zal waarschijnlijk op de beproefde manier via de NTG worden verspreid. De versie die daarna komt, zou misschien ook weer, met een nieuwe, duidelijke afspraak, bij de NTG onderdak kunnen vinden.

Verslag werkgroepen

WG Spelling: Erik Frambach vraagt namens de WG om hulp bij het verder corrigeren van de huidige woordenlijsten.

WG Nederlandse TeX: Werkgroep lid Johannes Braams zal zijn verslag later inleveren.

WG TeX & SGML Taco Hoekwater is eigenlijk te druk met andere werkzaamheden, en zoekt vervanging als contactpersoon voor deze WG.

Wat verder ter tafel komt Er ontstaat een discussie over de (gewenste) status van de WG educatie. Besloten wordt dat de WG Educatie niet langer het predicaat sluimerend zal hebben maar actief en zal worden bemand door Kees van der Laan, Piet van Oostrum en vermoedelijk Piet Tutelaers. De werkgroep zal in een eerste verslag aangeven welke activiteiten worden ondernomen.

4.11 Rondvraag en sluiting

Lustrum Hans Hagen roept leden op om zich te melden bij de lustrumcommissie, o.a. met ideeën voor de mogelijk 2-3 dagen beslaande NTG lustrumbijeenkomst. Thema zou kunnen zijn, iets als "Hoe maak je goede documenten hoe doe je dat in TeX?" Gedacht wordt aan een aantal prominente sprekers, o.a. op typografisch gebied. Taco Hoekwater meldt dat de MAPS voor deze gelegenheid extra dik (500 pagina's), breed en kleurig zal worden!

LUG CD René van der Heijden vraagt zich af of er iets is gekomen van het voornemen om van alle LUGS de publicaties op een CD-ROM te brengen. Erik Frambach meldt dat dit project slaapt. De NTG heeft als enige wel alles op Internet staan. Het gaat hier om maar liefst 80 MB aan documenten, in PDF, zoveel mogelijk met PS fonts.

Afsluiting Er volgden boeiende lezingen en in de pauzes was er veel te bekijken, te lezen en te bepraten (en voortreffelijk te eten en te drinken...) De laatste lezing van de dag bleek vooral een reeks voorzetten door de spreker, waarop een levendige discussie in de zaal volgde. De kenners op het gebied van SGML konden hierdoor hun meningen aan elkaar toetsen terwijl de argeloze toehoorder meer te weten kwam over "wat is SGML in hemelsnaam?". Een ontspannen en levendig besluit van deze dag, die door een kleinere groep nog geanimeerd werd voortgezet aan de dis van een gezellig eethuis waar de NTG een eigen eetkamer had gereserveerd.

Volgende bijeenkomst De leden worden uitgenodigd op de volgende vergadering, op zaterdag 15 november in het AMC te Amsterdam. Thema wordt "Hacken zonder hoogtevrees".

Bericht Van de voorzitter

Erik Frambach
Rijksuniversiteit Groningen
email: E.H.M.Frambach@eco.rug.nl

Malaise?

Een tendens die we al een tijdje gadeslaan is het dalen van het ledental van Local TeX User Groups. TUG heeft daar enorm onder geleden maar ook bij andere LUGs zijn de symptomen te zien. Een van de oorzaken die steeds wordt genoemd is dat men tegenwoordig via Internet alles kan vinden wat men nodig heeft, van programmatuur tot en met ondersteuning. Waarom zou je dan nog lid worden/blijven van een LUG? TUG gaat dit probleem als volgt te lijf. Geef alle leden elk jaar een complete set cdroms (TeX Live, 4allTeX, CTAN). Voor leden is het immers uiteindelijk veel gemakkelijker om vanaf cdrom te installeren dan moeizaam alle spullen over een dun telefoonlijntje binnen te halen, als ze al weten wat ze precies willen hebben. Verschillende andere LUGs verspreiden ook cdroms (met name NTG, DANTE en UKTUG) maar minder systematisch.

De vraag is natuurlijk hoe lang cdroms nog zo'n voordeel bieden. En meer fundamenteel: is verspreiding van software de belangrijkste taak van een LUG? Heeft een LUG dan verder niets wezenlijks te bieden? Als dat zo is zal het snel gedaan zijn met de LUGs. Een bezinning op de functie en de toekomst van LUGs, de NTG in het bijzonder, lijkt me geboden.

Maar er is ook positief nieuws. Een nieuwe LUG heeft zich gemeld. De Griekse TeX-gebruikers hebben zich verenigd in de *Greek TeX's Friends Group* (zie <http://obelix.ee.duth.gr/eft/> voor meer info).



Verder is er Friesland een clubje bezig met het ontwikkelen van een taaldefinitie voor BABEL. Ook Friese afbreekpatronen zouden er moeten komen. Zou een Friese TeX-gebruikersgroep levensvatbaar zijn? In Spanje hebben de Catalanen ook hun eigen club (zie [\[www-lsi.upc.es/~valiente/tug-catalan.html\]\(http://www-lsi.upc.es/~valiente/tug-catalan.html\)\) naast de gewone Spaanse TeX-gebruikersgroep \(zie <http://gordo.us.es/Activades/GUTH/>\)...](http://</p></div><div data-bbox=)

Sinds kort heeft ook de Russische TeX-gebruikersgroep CyrTUG haar eigen www-site: <http://www.cemi.rssi.ru/cyrtug/>. Netwerken is/zijn kennelijk erg belangrijk.

CyrTUG Ассоциация пользователей
кириллического TeX'a
(Cyrillic TeX Users Group)



English version [here](#)
Russian version (Win 1251) [here](#) (Анон-а-а-а-а-а)
Russian version (KOIS-R) [here](#) (+ΑΑΑ ΕΙΛΕΘΙ-ΕΑ)

Lustrum

Volgend jaar hoopt de NTG haar 10-jarig jubileum te vieren. Een (nog steeds te kleine!) lustrumcommissie is dat feest aan het voorbereiden. Bij die gelegenheid willen we aandacht schenken aan *typografie*. TeX heeft daarin lang ver voorop gelopen maar wordt tegenwoordig flink beconcurrereerd door kwalitatief gelijkwaardige producten. Qua programmeerbaarheid zie ik nog steeds geen belangrijke concurrentie, maar het oog wil ook wat. Gelikte presentaties en knap vormgegeven boeken en folders zijn gemeen goed geworden in onze steeds meer visueel gerichte samenleving. TeX heeft alles in huis om daar een belangrijke rol in te spelen, maar dan moeten we daar wel meer aandacht aan gaan schenken.

Prijsvraag

Over typografie gesproken. Het kan niemand ontgaan zijn dat de vormgeving van de MAPS geheel vernieuwd is. Het ziet er (naar mijn bescheiden mening) heel modern, strak en professioneel uit. De MAPS gaat met zijn tijd mee, en zo hoort dat ook. Heroverwegen, vernieuwen, oppoetsen en uitmesten is van tijd tot tijd nodig. In dit kader leek het ons zinnig om ook ons logo eens kritisch te bekijken. Wij hopen dat de prijsvraag naar een nieuw ontwerp iets fraais zal opleveren waar we weer 10 jaar mee vooruit kunnen.

Bericht Redactioneel

Taco Hoekwater

Voor U ligt al weer MAPS nummer 19. Na de toch wel beschamende hoeveelheid fouten en ongelukjes uit de vorige MAPS, voel ik me er niet gerust op. Daarom zal ik maar niet zeggen dat deze MAPS stukken beter is dan de vorige. Wat deze MAPS echter beslist *wel* is, is *anders*.

De afgelopen maanden is er door ons veel tijd besteed aan het her-overwegen van de layout. We zijn er zelf erg tevreden over, en we hopen maar dat het ook bij de lezers in de smaak valt. Anders is het zeker! Elders zult u diverse stukken vinden over het ontwerp en de implementatie van de nieuwe layout, en ook op de komende bijeenkomst hebben we wat ruimte gekregen om een en ander toe te lichten.

Met al die interne aandacht voor typografie, kon het niet uitblijven dat er ook diverse artikelen over typografische zaken gaan. Eerst en vooral wil ik daarbij de twee artikelen van Philip Taylor over boek-ontwerp noemen. Deze uitstekende artikelen, de eerste over de achtergronden van boekontwerp, de tweede toegespitst op toepassing binnen \TeX , geven samen een indruk van hoe nauw \TeX feitelijk aansluit bij het ouderwetse vak van *zetter*.

In deze MAPS echter ook aandacht voor de 'concurrentie': Frans Goddijn flirt met Quark Express, en Siep Kroonenberg geeft aan dat ze gebruik maakt van PageMaker. Ikzelf heb de nieuwste FrameMaker op mijn bureau staan om uitgepakt te worden ... ter compensatie laat Erik iets zien wat beslist niet kan met deze commerciële pakketten: het typesetten van Chinees blijkt binnen \TeX zelfs buiten China redelijk eenvoudig.

Natuurlijk is er ook ruimte voor een verslag van TUG '97. Het hoeft nauwelijks gezegd te worden dat de 'crazy dutchmen' zich uitstekend vermaakt hebben in San Francisco. Als reactie op het doorslaande succes dat Hans Hagen aldaar boekte met zijn presentatie van $\text{CON}\TeX$ is er in deze MAPS ook een flink stuk ruimte gereserveerd voor de eerste 25 paragrafen van 'CON \TeX voor beginners'.

Ook weer veel aandacht voor PostScript: in een levendig uitgebreid artikel laat Kees van der Laan zien dat het verrassend eenvoudig is om zelf te programmeren in PostScript. Maarten Gelderman gaat in de 'Toolbox' in op het gebruik van de diverse programma's rondom \TeX die vaak (ook) PostScript manipuleren.

Wat nog overblijft zijn de manuals: een voorbeeld van een 'local guide' van de hand van Maarten Gelderman, de gebruikers handleiding voor Kluwer's class file, en een installatie-handleiding voor 'em \TeX voor Windows'. De eerste geeft een houvast voor hoe een 'local guide' eruit zou kunnen zien en geeft terloops een inleiding in het gebruik van de JED tekst editor; de tweede laat een 'typisch' manual zien bij een van 'article' afgeleide klasse; en de derde helpt mensen op pad die moeite hebben met de vaak lastige installaties die nodig zijn voor em \TeX .

Er is gelukkig nog wat plaats en tijd om vast even stil te staan bij de dingen die komen gaan: het aanstaande lustrum van de NTG is al vlakbij. Natuurlijk kan de MAPS niet achterblijven bij het feestgedruis, en daarom zal de volgende MAPS extra extra dik en kleurrijk worden.

In ieder geval zal de 'Toolbox' terugkomen, het tweede deel van 'CON \TeX voor beginners', een artikel over fractals in PostScript, en we hopen Philip Taylor te kunnen strikken voor een artikel over 'verantwoord kleurgebruik'.

Verder verwachten we: recenties van True \TeX , MetaFog, KDVI versie 1.0, LyX versie 1.0, jade \TeX en 4 \TeX 4; verslagen van euro \TeX en EP '98; achtergrond-artikelen over een converter van SGML naar con \TeX t, de \LaTeX output-routine, en over de interne werking van de Kluwer classes; en uiteraard de uitslag van de ontwerpwedstrijd.

Maar dat is alles bij elkaar net genoeg voor een 'dikke' MAPS, en zeker niet voor een 'extra extra dikke', dus hierbij een dringende oproep voor artikelen ter plaatsing. Artikelen mogen over van alles gaan (zolang er een verband met \TeX is uiteraard), en we accepteren heus niet alleen 'TeXnische Wonderen', ook 'gewone' belevenissen kunnen bijzonder interessant zijn!

Namens de redactie veel leesplezier toegewenst!

De redactie van deze MAPS bestond uit: Gilbert van den Dobbelsteen, Siep Kroonenberg, Gerard van Nes, Jos Winnink en Taco Hoekwater. Oplage MAPS: 350.

Bericht Het weten waard

Taco Hoekwater

TeX kalender 1998

29–31 maart 1998	EuroTeX '98 Saint Malo, Frankrijk
1–3 april 1998	Electronic Publishing '98 Saint Malo, Frankrijk
17-18 april 1998	Multi-lingual Computing Cambridge, Groot Brittanië
17–21 augustus 1998	TUG '98 Torun, Polen

2 Glossary

Gebruikersgroepen

TUG	: TeX 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	: TeX and TUG News; TUG
TUGboat	: TUG
MAPS	: Minutes and AppendiceS; NTG

Diversen

CTAN	: Comprehensive TeX Archive Network; sites waar men 'anonymous ftp' kan ge- bruiken om TeX/LaTeX-achtig materiaal te verkrijgen. CTAN is de 'home' voor de officiële versie van LaTeX etc. CTAN sites zijn: ftp.dante.de, ftp.tex.ac.uk en pip.shsu.edu
FGBBS	: NTG's Bulletin Board
AllTeX	: TeX, LaTeX, etcTeX
ltxiii	: LaTeX 3.0
4TeX	: Het volledige TeX runtime systeem voor MS-DOS PC systeem, gebaseerd op emTeX en 4DOS.
4allTeX	: De 4TeX applicatie plus alle mogelijke gerelateerde files en utilities, gedistribu- eerd op de diskette set en CD-ROM.
AMS	: American Mathematical Society
SGML	: Standard Generalized Markup Language

3 NTG's TeX Bulletin Board System

Op het TeX Bulletin Board van de Nederlandstalige TeX Gebruikersgroep (FGBBS) is een zo volledig en actueel mogelijke TeX, emTeX, LaTeX, TEX-NL en MusicTeX 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 **026-3217041**.

4 NTG's winkel

Via de NTG is beschikbaar:

□ **4all \TeX dubbel cd-rom:**

Ruim 1.1 Gbyte (ruim 50.000 files) aan onder meer $\mathbb{4}\mathbb{T}\mathbb{E}\mathbb{X}$ utilities, fonts, $\mathbb{T}\mathbb{E}\mathbb{X}/\mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}/\mathbb{M}\mathbb{E}\mathbb{T}\mathbb{A}\mathbb{F}\mathbb{O}\mathbb{N}\mathbb{T}$ /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 $\mathbb{4}\mathbb{T}\mathbb{E}\mathbb{X}$ manual: *f* 60,-.

□ **$\mathbb{T}\mathbb{E}\mathbb{X}$ live cd-rom:**

Kant-en-klaar $\mathbb{T}\mathbb{E}\mathbb{X}$ systeem voor Linux, DEC, HP, IBM, NeXT, SGI, en SUN Unix systemen, alsook voor win32. Door Thomas Esser en Sebastian Rahtz, gebaseerd op Web2C van Karl Berry.

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

□ **Syllabus Advanced $\mathbb{T}\mathbb{E}\mathbb{X}$ 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 (La) $\mathbb{T}\mathbb{E}\mathbb{X}$ (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.

5 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.

6 De NTG najaarsbijeenkomst

De NTG najaarsbijeenkomst 1997 zal op zaterdag 15 november op het Academisch Medisch Centrum (AMC) te Amsterdam plaatsvinden. Het formele gedeelte van de vergadering zal ongeveer een uur in beslag nemen.

7 MAPS 98.1

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

1 april '98 (MAPS 98.1; #20)

1 oktober '98 (MAPS 98.2; #21)

Aanleveren kopij voor de komende MAPS:

- *Bij voorkeur* in $\LaTeX 2_{\epsilon}$ gebruikmakend van de `maps.cls`. Deze stijlfile is via de redactie te verkrijgen en beschikbaar op de TEX-NL fileserver, `archive.cs.ruu.nl` (ftp-site) en FGBBS (026-3217041).
- Daarnaast kunnen bijdragen ingestuurd worden gemaakt met `ltugboat.sty` of `article.sty` / `report.sty`.
- Verder zijn bijdragen vanzelfsprekend ook welkom in *plain-TeX* of ongeformatteerd.
- Plaatjes bij voorkeur als (Encapsulated) PostScript file.

Daar MAPS bijdragen in plain TeX altijd worden omgezet naar \LaTeX , verdient vanzelfsprekend aanbidding van materiaal in \LaTeX 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.

Bericht Begroting NTG 1998

Wybo Dekker

In tabel hieronder vindt U de voorlopige begroting voor 1998 van de Nederlandstalige T_EX gebruikersgroep. Een toelichting volgt na de tabel.

	Inkomsten	Uitgaven
Contributie	24000	
Rente	1000	
Verkoop 4allTeX CD	PM	
Kantoorkosten		1300
Drukkosten		12000
Porti		4500
Computerfaciliteiten		400
Bijeenkomsten		900
Zaalhuur		250
Bestuurskosten		300
Representatie		1000
Bijdrage Bursary fund		1000
Sponsoring Eurotex		1000
Sponsoring FGBBS		1000
Onvoorzien		1350
	25000	25000

Contributie De post contributie is gebaseerd op het aantal betalende leden in oktober 1997:

33	instituten	×	245 =	<i>f</i> 8085
4	aanvulling	×	65 =	260
8	studenten	×	60 =	480
115	NTG-ers	×	90 =	10350
58	NTG+TUG-ers	×	81 =	4698
				<i>f</i> 23873

Rente De vereniging heeft in de afgelopen jaren een behoorlijk kapitaal opgebouwd (circa *f*30000). Als dit niet nodig is om een tegenvaller op te vangen is het mogelijk behoorlijk wat rente te krijgen.

Verkoop 4allTeX CD De opbrengsten die van deze CD verwacht kunnen worden zijn nog niet bekend.

Kantoorkosten Dit is bedoeld voor materiaal voor de secretaris en penningmeester. De hoogte is bepaald aan de hand van de hoogte van de uitkomst over de eerste drie kwartalen van 1997. Verder de kosten van Kamer van Koophandel

Drukkosten Kosten voor het drukken van MAPS, nieuwsbrieven en dergelijke.

Porti Kosten voor verzending van MAPS, nieuwsbrieven, facturen en dergelijke.

Computerfaciliteiten Hiervan worden incidentele voorzieningen voor het bestuur betaald. Verder de kosten voor de internet pagina en domeinnaam registratie.

Bijeenkomsten Kosten voor koffie, thee en lunch voor deelnemers aan de bijeenkomsten.

Zaalhuur Kosten voor zaalhuur voor bestuursvergaderingen en bijeenkomsten.

Bestuurskosten Hieronder vallen kosten als telefonische vergaderingen, vergoeding reiskosten voor een eventuele fysieke bijeenkomst etc.

Representatie Als bestuursleden van zusterverenigingen bij onze bijeenkomst uitgenodigd worden, kan een tegemoetkoming in de kosten worden gegeven. Verder voor aankoop van prijzen, diners voor sprekers op bijeenkomsten en dergelijke.

Bijdrage Bursary fund Het bestuur stelt voor om jaarlijks een bijdrage te leveren aan het Bursary fund van TUG. Dit fonds is bedoeld om T_EX gebruikers met financiële problemen in de gelegenheid te stellen een TUG bijeenkomst bij te wonen.

Sponsoring EuroTeX Het bestuur is van plan vanaf 1998 een bijdrage te doen ten behoeve van een dan in te stellen jaarlijkse "EuroTeX-Award".

Sponsoring FGBBS De jaarlijkse bijdrage aan Frans Goddijn's BBS.

Bericht Ontwerpwedstrijd



De NTG gebruikt nu al sinds jaar en dag het logo dat gebruik maakt van de calligrafische letter uit de Computer Modern math fonts, zoals hierboven afgebeeld.

Ondertussen is dat logo zo verouderd dat het zaak is de oude metafont bestanden voorzichtig te backuppen omdat de gebruikte sources niet meer op CTAN staan (Knuth's veranderingen in 1992 hebben onder andere de \mathcal{T} aangepast).

Onder andere daarom, maar ook omdat we behoefte hebben aan een meer 'eigen' stijl, zijn we op zoek naar een nieuw ontwerp voor het logo en de briefstijl.

Wat we nodig hebben zijn de volgende zaken:

1. Een logo in breed formaat. Tussen 3-5 tekstregels hoog, over volle breedte, om te gebruiken op de eerste pagina van de verslagen en in het briefhoofd. Binnen het ontwerp moet rekening gehouden worden met eventueel te plaatsen gegevens zoals adressen/girorekeningnummers, en het logo moet redelijk integreren met zowel een artikel in de maps als met een standaard ISO briefhoofd zoals de \LaTeX NTG-classes dat maken.
2. Een kleinere versie van het logo voor gebruik op vervolgpagina's en op andere plekken (bijvoorbeeld op naamkaartjes of memo-blaadjes). Liever niet over de volledige breedte, maar een 'echt' logo.
3. Een opzet voor een huisstijl voor de NTG. We bedoelen dan een set 'regels' die ervoor zorgen dat alle publicaties van de NTG direct als zodanig herkenbaar zijn en overkomen als een eenheid. Het zou natuurlijk leuk zijn als de integratie van de nieuwe MAPS (97-2) en de nieuwe WWW pagina's (<http://www.ntg.nl>) hierdoor verbeterd zou worden.

Wat we aanbieden aan de winnaar is naar keuze een geheel verzorgde reis naar TUG '98 of euro \TeX '98. Indien mocht blijken dat de winnaar toch al naar een van beide bijeenkomsten zou gaan, dan kan ook gekozen worden voor een financiële donatie van $f 100,- \times 10 \text{ jaar} = f 1.000,-$.

U kunt meedoen door een ontwerp toe te zenden aan het postadres van de NTG, onder vermelding van 'ontwerp-

wedstrijd'. Inzendingen dienen binnen te zijn voor 1 januari '98.

De kleine lettertjes:

Deelnemers moeten lid zijn van de NTG op het moment van inzending.

Inzendingen en copyright vervallen aan de NTG.

We behouden ons het recht voor aanpassingen te maken waar wij dat nodig vinden.

Over de uitslag kan niet worden gecorrespondeerd.

Indien de jury van mening is dat geen van de inzendingen voldoet, zal de prijs niet worden uitgereikt.

De technische eisen voor het logo zijn als volgt:

- Het bestandsformaat moet EPS (Encapsulated Post-Script) zijn, danwel door \TeX gezette tekst, danwel een combinatie van beide.
- Het logo moet acceptabel te printen zijn bij een minimale resolutie van 300 dpi + resolutie-enhancement (cf. uitvoer zoals verkregen uit een standaard hp Deskjet of Laserjet III).
- Het logo moet schaalbaar zijn naar een willekeurige grootte zonder kwaliteitsverlies, en zonder dat het logo opnieuw aangemaakt moet worden (dwz: geen bitmaps).
- Gebruik van een steunkleur of overloop-tinten wordt aangemoedigd, maar in dat geval is er tevens een grijs logo nodig met maximaal 4 discrete grijswaarden (dit zodat er altijd een duidelijk verschil is tussen de grijstinten, zelfs op een Deskjet of Laserjet III, waarop overlooptinten ontzettend lelijk worden).
- Gebruik van commerciële fonts is toegestaan, maar we hebben een exacte specificatie van de gebruikte fonts nodig.

De technische eisen voor de huisstijl zijn als volgt:

- Gebruik van een steunkleur of overloop-tinten: zie hierboven.
- Voor de fonts: naar keuze, maar de voorkeur gaat uit naar Outline-fonts die vrij gebruikt kunnen worden.

Veel succes! De (uiteraard) deskundige jury wacht met spanning af.

Het NTG bestuur.

Bericht 4allT_EX versie 4

Erik Frambach
Rijksuniversiteit Groningen
email: E.H.M.Frambach@eco.rug.nl

abstract

De vierde editie van de 4allT_EX cdroms komt eraan. Dit artikel beschrijft wat er zoal nieuw is in deze editie.

keywords

4T_EX, 4allT_EX, EMT_EX, plug & play, cdrom

Introductie

4allT_EX is een kant-en-klaar systeem voor MS-DOS-, WINDOWS- en OS/2-gebruikers. Sinds 1994 is het systeem op cdrom verkrijgbaar. Het was destijds de eerste T_EX cdrom die werkelijk ‘plug & play’ T_EX mogelijk maakte. Na die eerste revolutionaire cdrom zijn twee nieuwe edities uitgebracht, waarvan in totaal meer dan 5000 exemplaren wereldwijd zijn verkocht, voornamelijk van de derde editie uit 1995.

Een nieuwe editie

Sinds 1995 is er heel wat veranderd in de MS-DOS- en WINDOWS-wereld. Vandaar dat het ons (Wietse Dol, Maarten van der Vlerk en mij) zinvol leek een nieuwe editie uit te brengen die weer helemaal up-to-date zou zijn. Een belangrijk aspect daarbij was dat we dankzij de opkomst van WINDOWS 95 en WINDOWS NT een aantal faciliteiten konden toevoegen die voorheen onmogelijk waren.

Wat is er nieuw

Om te beginnen moest natuurlijk van alle relevante software de nieuwste/beste versie op de cdrom komen. Wat betreft de T_EX/L^AT_EX-kant hebben we dankbaar gebruik gemaakt van de T_EX Live cdrom die door Sebastian Rahtz is geproduceerd.

Daarin hebben we foutjes verbeterd en pakketten ge-update. Daarnaast hebben we er een aantal zaken aan toegevoegd die wij belangrijk achten.

De belangrijkste verschillen met de derde editie van 4allT_EX zijn:

- Bij de installatie wordt nu onderscheid gemaakt naar het besturingssysteem waarop 4T_EX gaat draaien. Dat maakt het mogelijk om verschillende configuraties naast elkaar te hebben die toch allemaal dezelfde sources gebruiken. Bovendien kan per besturingssysteem optimaal gebruik gemaakt worden van de faciliteiten die dat systeem biedt. Dat geldt met name WINDOWS 95 en WINDOWS NT waarmee aanzienlijk meer mogelijk is dan voorheen.
- 4T_EX onder WINDOWS 95/NT kan programma's parallel starten, zodat het menu beschikbaar blijft naast bij voorbeeld een WINDOWS editor en een WINDOWS previewer. 4T_EX is ook in staat om WINDOWSprogramma's die uitsluitend binnen 4T_EX zinvol zijn automatisch af te sluiten als 4T_EX zelf afsluit (“kill”).
- 4T_EX ondersteunt nu nog uitsluitend computers met een 80386 cpu of beter. Wie een 80286 of minder heeft zal zich met de standaard EMT_EX spullen moeten redden.
- De spell-checker (nog steeds AMSPELL vanwege problemen met ISPELL) ondersteunt nu ook Zuidafrikaans, en vanzelfsprekend ook de nieuwe Nederlandse spelling.
- Er worden nu veel meer verschillende editors meegeleverd en ondersteund. Onder andere de hele TSE-familie (voorheen QEDIT), PFE, WINEDT, NOTEPAD, EMACS, ZEUS, QUICKEDIT (niet te verwarren met Qedit of Quickeditor) en MULTI-EDIT.
- 4T_EX draait nu op de nieuwste versie van 4DOS, 6.0, en maakt gebruik van een aantal nieuwe handigheden daarvan.
- Chinees wordt nu ondersteund via het L^AT_EX-package CJK. Fonts en alles eromheen is klaar voor gebruik.
- Op verzoek van DEK *himsel*f kan 4T_EX nu ook direct interactief T_EX draaien (de “T_EX prompt”) door als main file null op te geven.
- Vanwege verschillende dvi-drivers die niet met virtuele fonts overweg kunnen is het “devirtualiseren” met behulp van DVICOPY beter in het systeem geïntegreerd. Onder andere voor DVIWIN is dat essentieel als je bij voorbeeld met Times fonts werkt. Ook voor DVISCR is het prettig want ook die raakt snel in geheugenproblemen bij grote aantallen virtuele fonts.
- Nieuw is ook dat documentatie in HTML voor ook voor DOS-gebruikers grafisch (dus met plaatjes) kan worden weergegeven door de browser Arachne. Een hele verbetering in vergelijking met DOSLYNX.

- WINDOWS-gebruikers kunnen natuurlijk NETSCAPE of INTERNET EXPLORER kiezen.
- Het METAPOST-menu is sterk verbeterd. GHOSTSCRIPT (DOS- of WINDOWS-versie kunnen gebruikt worden om plaatjes te previewen.
 - De lijst van ondersteunde printers is nog verder uitgebreid. Geheel nieuw is de faciliteit om via GHOSTSCRIPT PDF te genereren.
 - Ook 1200 dpi laserjets worden nu ondersteund dankzij een 32-bits versie van DVIHPLJ.
 - De POSTSCRIPT TYPE 1 versies van de Computer Modern en AMS fonts die recentelijk is vrijgegeven wordt uiteraard gebruikt.
 - Conversie van *any*TeX naar HTML met behulp van Gurari's TeX4HT wordt nu ondersteund.
 - Het CONTeXt format van Hans Hagen (Pragma) is beschikbaar, inclusief documentatie.
 - GNUPLOT (zowel in DOS- als WINDOWS-uitvoering) is direct toegankelijk.
 - GSVIEW, de WINDOWS-versie van GHOSTSCRIPT, kan nu als viewer gekozen worden.
 - Een recente L^ATeX 2_ε WINDOWS help file kan geraadpleegd worden voor informatie over L^ATeX-commando's.
 - Met behulp van L^ATeXMACis het typen van onder andere formules een stuk eenvoudiger door een WYSIWYG-interface.
 - Het WINDOWS-programma L^ATeXCAD kan gebruikt worden om tekeningen te maken die gemakkelijk met TeX of L^ATeX zijn te gebruiken.
 - 4TeX is door verschillende optimalisaties een stuk sneller geworden. Intern zijn grote stukken code herzien en vereenvoudigd. Extern is enorme snelheidswinst (tot een factor 2!) geboekt door de EMTeX compiler zo snel mogelijk toegang tot bestanden te geven met een minimum aan subdirectories (subdirectory searching vreet tijd, vooral op relatief langzame cdroms).
 - Dankzij GHOSTSCRIPT is nu van alle 35 standaard POSTSCRIPT fonts een *public domain* versie beschikbaar.
 - Beide cdroms zijn met een WWW browser te 'browsen' dankzij een file `index.htm` in elke directory, met verwijzingen naar andere directories en files.
 - In de root staan bovendien complete lijsten van alles wat er op de cdrom te vinden is (à la FILES.byname op CTAN). In de EMTeX root staat een file `ls-R` die snelle toegang tot alle sources op de cdrom mogelijk moet maken voor WEB2C-gebruikers.
 - De nieuwe cdroms zijn houden zich aan de ISO-9660 standaard waardoor ze op vrijwel elk type computer/-besturingssysteem leesbaar zijn. Dat is handig want het grootste deel van alle TeX-spullen is op willekeurige systemen te gebruiken. WINDOWS-gebruikers kunnen dankzij de 'Joliet'-uitbreiding ook lange bestandsnamen zien. Met name op de tweede cdrom komen die voor, onder andere bij WWW-bestanden met als extensie `html`.
 - De complete documentatie van 4TeX staat in elektronische versies (POSTSCRIPT en PDF) op de cdrom. Uiteraard is de documentatie volledig bijgewerkt. Het installeren van 4TeX wordt nu veel uitvoeriger beschreven.
 - Het installatiescript is aanzienlijk verbeterd. Het is nu ook veel gemakkelijker om "alles" op de harde schijf te installeren, waarbij eenvoudig gekozen kan worden voor onderdelen. Ook *na* de installatie kunnen later eenvoudig onderdelen toegevoegd worden zonder het installatiescript weer door te moeten ploegen.
 - Op de tweede cdrom is wederom een keur aan programmatuur te vinden die voor TeX-gebruikers interessant kan zijn. Verschillende distributies van andere TeX user groups is voorhanden, evenals andere 'shells' zoals EMTeXGI en TeXTELMEXTEL.
 - De complete 'L^ATeX catalogue' is er in HTML, POSTSCRIPT en PDF. De catalogue beschrijft een enorme lijst L^ATeX-packages.
 - Informatie over usergroups is natuurlijk zo compleet mogelijk. De nieuwe Griekse gebruikersgroep is toegevoegd. Bij verschillende andere groepen zijn vele elektronische tijdschriften toegevoegd.
 - De TeX-cursussen zijn ge-update en uitgebreid. Zo is er nu ook een Spaanse en een Griekse cursus.

Met dank aan

Ook aan deze nieuwe versie hebben weer heel veel mensen bijdragen geleverd door software te leveren, hints te geven voor verbeteringen en uitbreidingen, bug-reports, en beta-tests. Te veel namen om hier te noemen, dus aan allen: hartelijk dank!

Verkrijgbaarheid

Net als voorheen zal ook deze editie bij de NTG te koop zijn. De prijs voor de dubbel-cdrom inclusief de gedrukte handleiding blijft net als vroeger 60 gulden. Nieuw is dat de cdroms ook zonder boekje te koop is, voor 30 gulden. Prijzen zijn inclusief verzending.

Via de bekende kanalen zoals WWW, mailing lists en Usenet zal spoedig nadere informatie bekend worden gemaakt.

NTG

NTG-leden hebben zoals gewoonlijk een streepje voor. Zij krijgen de cdroms *gratis* thuisgestuurd. De handleiding kunnen voor ze slechts 15 gulden aanschaffen. Als dat geen goede reden is om NTG-lid te worden...

Bijlage 1

Het herontwerp van de MAPS

Siep Kroonenberg
n.s.kroonenberg@eco.rug.nl

Deze zomer heeft de MAPS redactie gediscussieerd over een nieuwe vormgeving. De mogelijkheden zijn de laatste jaren enorm toegenomen en daarvan zouden we best wat mogen terugzien in de MAPS.

Papierformaat

Nu bijvoorbeeld de MAPS wordt ge-offset kunnen we afwijken van het aloude A4 formaat. In de discussie over de nieuwe opmaak is even gespeeld met de mogelijkheid van een veel kleiner formaat, met een éénkoloms-opmaak. Dat zou het leven aanzienlijk vereenvoudigen, maar uiteindelijk besloten we toch twee kolommen te handhaven, zowel uit esthetische voorkeur als omdat het veel papier bespaart. Wel is het formaat nu korter en iets smaller.

Fonts

We hebben ook gekozen voor een tweede font-familie, voor gebruik in o.a. koppen en bijschriften. Deze tweede fontfamilie is Adobe's versie van Frutiger. Tevens fungeert Frutiger-Bold buiten math mode als vervanger voor Times-Bold. Frutiger is iets verkleind ten opzichte van Times om ze beter met elkaar te laten harmoniëren.

Na rijp beraad hebben we afgezien van een ander tekstfont: we verwachtten dat dat teveel technische problemen met zich mee zou brengen.

Het is nogal bewerkelijk om fonts te installeren in \TeX . Daarom bieden we auteurs de optie van een vereenvoudigde font setup met de standaardfonts Times en Helvetica.

Typografische parameters

Vrijwel alle opmaak-parameters moeten speciaal afgestemd worden op een twee-kolomsopmaak. Om te beginnen is de tekstkolom veel smaller. Om niet nog meer van de breedte af te snoepen moet er dus niet teveel worden ingesprongen, dwz. niet te ver en niet te vaak. 'Niet te ver' was een kwestie van parameters veranderen; vanwege 'niet te vaak' hebben we enkele list-achtige structuren geïntroduceerd die helemaal niet inspringen; inplaats van inspringen wordt er een regel overgeslagen.

Uitvullen gaat ook slechter naarmate de tekstkolom smaller wordt. Daarom is uitvullen binnen list-structuren uitgezet.

Een smallere kolom maakt het aan de andere kant mogelijk om met een kleinere letter en krappere regelafstand te werken: die zijn op respectievelijk 9,5 en 11 punt ingesteld.

Een twee-kolomsopmaak kan makkelijk rommelig en onrustig worden, ook al omdat de bladspiegel zo vol is. Een populaire methode om orde te scheppen is het laten lijnen van regels in beide kolommen. Voor een deel kan dit tot stand worden gebracht door de verticale maten goed te kiezen, maar er zijn altijd elementen die het stramien verstoren. Dus is er een mechanisme nodig om daarna weer terug te keren tot het stramien. \TeX leent zich slecht tot dit soort rekenpartijen, maar Taco heeft er wat op gevonden.

De aanhef van bijlagen was een mooie gelegenheid om even uit het gareel te lopen, door gebruik te maken van het tweede font en door te spelen met de uitlijning.

Een desktop publishing programma (PageMaker) heeft goede diensten bewezen bij het snel uittesten van verschillende fontfamilies en bladindelingen.

Eénkoloms opmaak

Er zijn altijd bijdragen die een bredere kolom nodig hebben. De bladspiegel van de éénkoloms-variant is $\frac{3}{4}$ van de standaard bladspiegel. Zonde van al dat lege oppervlak, maar regels over de volle breedte lezen nu eenmaal niet lekker. We hebben de mogelijkheid gecreëerd om voor b.v. verbatim de volle breedte te gebruiken, en om marginal notes in de brede buitenkantlijn te plaatsen.

Paginaheaders- en footers zijn identiek aan die van de twee-kolomsopmaak, althans wat betreft uiterlijk. We hopen dat dat voldoende is om de eenheid te bewaren.

Tenslotte

Op het moment dat ik dit schrijf weet ik nog niet hoe het zal uitpakken voor de MAPS in zijn geheel: de stijlfile is op dit moment alleen uitgetest op een aantal oude MAPS artikelen. Ongetwijfeld zal er de komende nummers nog één en ander worden opgepoetst aan de nieuwe MAPS stijl. Natuurlijk zijn we benieuwd wat de lezers ervan zullen vinden: commentaar is welkom.

Bijlage 2

De macros uit de nieuwe maps class

Taco Hoekwater

abstract

Dit artikel beschrijft de macros die gebruikt worden in de nieuwe MAPS class file. De nadruk ligt op de macros die proberen een stramien te ondersteunen

keywords

maps class file, macro design, stramien

1 Hoe en waarom

Wie kijkt naar de bestaande classes en stylefiles voor \LaTeX , zal al snel tot de conclusie komen dat betreffende layouts eerder in elkaar geflansd zijn door een goed bedoelende amateur dan *ontworpen* door een professionele kracht. Vaak is dat vooral te zien aan slordig bij elkaar gezochte fonts, relatief te grote sectie-kopjes, en lelijke kop- en voetregels.

Ook de MAPS stylefile leed aan datzelfde probleem, met name doordat er door verschillende mensen aan geknutseld was in de loop der jaren. Bovendien was de stylefile die gebruikt werd toch al aan vervanging toe omdat het nog een oud latex2.09 bestand was, en als nieuwe hoofdredakteur wil je je natuurlijk ook een beetje afzetten tegen je voorgangers.

Na overleg binnen de redactie hebben we Siep Kroonenberg gevraagd te komen met een nieuw ontwerp, met als belangrijkste eis dat die er ‘professioneler’ uit zou zien. Siep vertelt in een andere bijlage van deze MAPS over de design-beslissingen die kwamen kijken ontwikkeling van dat nieuwe class file; dit artikel gaat het hebben over de \TeX -macros die nodig waren.

Met het voorstel voor een nieuwe layout, waar wij allemaal erg tevreden mee waren, begonnen de problemen van de implementatie. Het is één ding om te zeggen hoe je het wilt hebben, maar er voor zorgen dat dat ook min of meer automatisch gaat is weer heel wat anders. . .

- Het nieuwe class file heeft een onderliggend stramien van basislijnen (53 in totaal), waar alle `\baselines` mee moeten lijnen.
- Er wordt op een nogal geavanceerde manier gebruik gemaakt van het NFSS. Met dank aan Siep kunnen we bijvoorbeeld ‘in het jaar 1997’ invoeren als ‘in het jaar 1997’, waarbij de hangende cijfers

vanzelf gaan, en komt er in de hele MAPS geen enkele **Times Bold** voor, terwijl we toch gewoon `\textbf` kunnen gebruiken.

- De lijst-omgevingen wijken sterk af van de standaard latex lijsten zoals ‘itemize’ en ‘enumerate’, om het nog maar niet over het begin van het artikel te hebben
- Zoals Hans Hagen al aangaf in zijn artikel in MAPS 16, lenen zijn macros voor verbatims zich ook voor gebruik van verschillende fonts. Daarom heeft het nieuwe class file de mogelijkheid om

a) Hans’ macros te gebruiken in plaats van het standaard ‘verbatim’ package

en

b) Hebben we wat code toegevoegd zodat de omgeving ‘prettyverbatim’ gebruik maakt van verschillende fonts.

Bijkomend voordeel: de macros van Hans zijn zowel stabiel als flexibeler als het ‘verbatim’ package.

- Grote blokken code voor tabellen en figuren zijn geleend van de class files van Kluwer Academic.

Dit artikel zal zich vooral concentreren op het eerste punt. Niet dat de andere punten zoveel moeilijker of makkelijker waren, maar omdat ik denk dat deze macros het interessantst zijn.

2 Het gebruik van een stramien

Een eerste probleem bij het hebben van een stramien is dat je het zou moeten kunnen zien in de uitvoer. Op die manier is er tenminste controle mogelijk of het allemaal wel netjes werkt. Het wordt dan immers een stuk makkelijker om aan te wijzen welke macro de schuldige is als de kolommen weer eens niet uitlijnen.

Daarom gaan de eerste macros over het zetten van het stramien zelf. Het effect van deze macros is te zien op de volgende pagina, terwijl de macros hieronder gepresenteerd worden.¹

Het eerste probleem is natuurlijk ervoor zorgen dat het stramien zelf op een vaste plek op de pagina staat. Er komt feitelijk maar één plaats in aanmerking, namelijk de linkerhoek van de kopregel. Helaas heeft \LaTeX geen

1. De meeste macros die hier worden afgedrukt zijn ietwat aangepast. Enerzijds om te zorgen dat ze passen binnen de nauwe kolommen van de MAPS, anderzijds om de gebruikte logica zo duidelijk mogelijk te maken

nette manier om daar iets aan op te hangen, dus het moet op de volgende manier, binnen de definitie van elke \pagestyle: ²	<code>\hbox to 0pt{% \vbox to 0pt{% \kern \topskip \kern \headsep</code>	
<code>\def\ps@empty{% \let\@mkboth\@gobbletwo \def\@oddhead{\@stramien\hss}% \let\@evenhead\@oddhead \let\@oddfoot\@empty \let\@evenfoot\@empty}</code>	<code>\@stramien\vss }% \hss }}</code>	5
Het gaat hier natuurlijk om de toevoeging van \@stramien. Dit is de eenvoudigste stijl die mogelijk is, 'headings' bijvoorbeeld ziet er als volgt uit:	Volgens de LaTeX Companion (pagina 85) is de afstand tussen de baseline van de kopregel en de eerste baseline van de tekst gelijk aan \headsep, maar de redenatie daar is niet al te duidelijk, en klaarblijkelijk moest ook \topskip nog toegevoegd worden. Het begin van deze pagina ziet er daarmee uit zoals aan gegeven in figuur 1.	10
<code>\def\ps@headings{% \def\@oddfoot{\sffamily\inf@tr\hfil\thepage}% \def\@evenfoot{\sffamily\thepage\hfil\inf@t1}% \def\@oddhead{\@stramien\sffamily\rightmark \hfil Bijlage \the@bijlagenummer}% \def\@evenhead{\@stramien\sffamily Bijlage \the@bijlagenummer\hfil\leftmark}% \let\@mkboth\@gobbletwo \let\sectionmark\@gobble \let\subsectionmark\@gobble}</code>	<code>\def\@stramien{\vbox{\hsize=\textwidth \strrules}} \def\@strrules{% @tempcntb\@stramienlines \advance\@tempcntb1 \loop \ifnum \strcount < \@tempcntb \global\advance \strcount 1 \repeat } \def\@stramienlines{53}% Kortom: 53 regels met een lijn erop. Wat overblijft is de definitie van van \@strrule. Even een paar handigheidjes: \def\thestrcount{{\scriptsize \number\strcount}} \def\@strnumber{\smash{% \raise \normalbaselineskip \hbox to \hsize{% \hss \rlap{`\ \thestrcount}% }}% \def\@stramien{\global \let\@stramien\@istramien} \def\@istramien{\hrule \@width #1\hsize \@height.1pt \@depth .1pt \kern -.2pt \kern \normalbaselineskip }</code>	15
Hieraan valt te zien dat er een paar eisen aan de geëxpandeerde macro \@stramien zitten:		20
<input type="checkbox"/> De macro moet een natuurlijke breedte van 0pt hebben, anders zouden de andere gegevens naar rechts opschuiven.		25
<input type="checkbox"/> De macro moet ook een diepte van nul hebben, want anders eindigt de rest van de pagina beneden het papier.		25
Nu definiëren we de macros \@stramien (die op zijn beurt \@stramien zijn uiteindelijke betekenis geeft) en \nostramien (die ervoor zorgt dat de lijnen weer verdwijnen).		30
<code>\def\@stramien{} \def\@stramien{\global \let\@stramien\@istramien} \def\@nostramien{\gdef\@stramien{}} \newcount\strcount</code>		35
We gaan de counter straks gebruiken om regels te tellen.		40
Nu de verticale plaatsing. De broodtekst moet op de lijnen vallen, en de afstand tussen de kopregel en de eerste baseline is meestal niet precies een geheel aantal \baselineskips, dus we moeten het wat netter uitrekenen:	De \kern -.2pt is om het effect dat de hoogte van de \hrule zelf heeft weer ongedaan te maken.	45
<code>\def\@istramien{% \global\strcount = 1</code>	2. Binnen latex definiëren de \ps@xxxxx commando's de verschillende stijlen voor de kop- en voetregels die worden aangeroepen met het commando \pagestylexxxxx. Op hun beurt definiëren deze \ps@xxxxx macros de macros die uiteindelijk door de output-routine gebruikt worden om de kop- en voetregels te zetten.	50

```

..\vbox(11.0+0.0)x455.24408, glue.... % dit is de hele kopregel
...\glue 0.0 plus 1.0fil
...\hbox(6.73709+1.94931)x455.24408
...\hbox(6.73709+1.94931)x455.24408, glue.... % dit de tekst in de kop
.....\hbox(0.0+0.0)x0.0, glue.... % \hbox to 0pt
.....\vbox(0.0+0.0)x466.62239, glue.... % \vbox to 0pt
.....\kern 10.0 % \kern \topskip
.....\kern 25.0 % \kern \headsep
.....\vbox(583.00081+0.0)x466.62239 % @@stramien ...
.....\rule(0.1+0.1)x455.24408
.....
.....\glue 0.0 plus 1.0fil minus 1.0fil
.....\OT1/pfrl/m/n/9.5 B % Bijlage ...
.....
..\glue 25.0 % \headsep
..\vbox(583.0+0.0)x455.24408
.....
.....\glue(\topskip) 3.51161 % en dit is \topskip:
.....\hbox(6.48839+2.05667)x212.62204, glue....% 3.51161 + 6.48839 = 10pt

```

Figuur 1.

`\@@strrule` roept de twee macro's hierboven aan, en dan nog wat grapjes om regelnummers in de kantlijn te zetten. De `\divide` / `\multiply` truc is de manier om in \TeX modulo-berekeningen te doen.

```

\def\@@strrule{%
\@tempcnta\strcount
\divide \@tempcnta 10
\multiply \@tempcnta 10
\ifnum \@tempcnta = \strcount
\@internalrule{1.040}%
\@strnumber
\else
\@tempcnta\strcount
\divide \@tempcnta 5
\multiply \@tempcnta 5
\ifnum \@tempcnta = \strcount
\@internalrule{1.020}%
\@strnumber
\else
\@internalrule1
\fi
\fi
}

```

Aan deze macro's valt beslist nog het een en ander te verbeteren (zoals de toevoeging van een gebruikers-interface en wat meer opties), maar het voldoet op dit moment heel aardig. Het zal wel nooit een apart package worden, want het toevoegen van `\@stramien` in de verschillende `\ps@xxxxx` commando's kan niet gedaan worden zonder hulp van degene die de class ontworpen heeft. Het alter-

natief – zelf de outputroutine aanpassen – is al helemaal uit den boze.

3 De problemen die \TeX heeft met vaste basislijnen

\TeX is nooit ontworpen om gebruikt te worden met een stramien zoals dit, hetgeen blijkt uit alle afwijkingen die standaard voorkomen op een pagina. En dan gaat het niet eens alleen over verticale `\skips`, maar ook:

1. Regels die hoger zijn dan een standaard regel zorgen ervoor dat `\lineskip` wordt toegevoegd. Hierdoor wordt de regelafstand ter plekke groter, en dit komt nooit meer goed. Grote boosdoener: in-lijn formules en voetnoot-tekens.
2. Verschillen in fontgrootte veranderen de baseline uiteraard, maar het is erg moeilijk om weer 'terug te vallen'.
3. Een heleboel 'standaard' glue's moeten aangepast worden zodat ze een discrete rij mogelijkheden hebben, en geen rek of krimp componenten.
4. Display formules hebben hoogte+diepte afgeleid van de natuurlijke maten met toevoeging van de verschillende `\displayskips`, maar de natuurlijke maten zijn niet makkelijk op te vragen
5. Hetzelfde geldt voor alle `\inserts` en `\aligns` (onder andere dus latex's 'tabular').
6. De witruimte rond `\section`-achtige commando's is slecht te voorspellen, en dan met name in de buurt van een pagina-breuk.

Deze problemen zijn nog redelijk op te lossen, gesteld dat er gebruik wordt gemaakt van een relatief simpel format, maar bij zoiets groots als latex wordt het al snel bijna onmogelijk om het goed te doen.

Voor punt 1. is bijvoorbeeld een redelijk elegante oplossing mogelijk in plain \TeX :

```
\baselineskip=0pt
\lineskiplimit=-\maxdimen
\lineskip=12pt
```

Simpel en doeltreffend, maar binnen een complex format als latex is het beslist niet veilig om zomaar even `\baselineskip` op nul te zetten!

4 Een aanzet tot een gedeeltelijke oplossing

4.1 Regelaafstand

Eigenlijk valt er binnen latex maar erg weinig te doen. Het enige wat ik kon bedenken was:

```
\def\imath{\smash\bgroup$}
\def\endimath{$\egroup }
\let\(\imath
\let\)\endimath
```

Helaas gebruikt niemand(!) de combinatie `\(\)` voor inlijn wiskunde.

4.2 Font groottes

Hier is heel weinig aan te doen, en er zit niet veel anders op dan overall waar de baselines afwijken van de ‘standaard’ met de hand in te grijpen.

De enige situatie die we kunnen automatiseren is dat de `\baselineskip` van `\large` meestal wel gelijk gemaakt kan worden met die van `\normalsize`. Voor `\small` zou het nog mogelijk zijn een oogje dicht te knijpen, maar de andere groottes zien er onzinnig uit.

4.3 Glues en parameters

In ieder geval:

```
\parskip 0pt
```

Want `\parskip` heeft normaal gesproken een rek- component. En uiteraard moeten ook alle `\baselineskips` niet kunnen rekken of krimpen.

Ook een hele verzameling commando’s die de witruimte rond lijsten regelen dienen geherdefinieerd te worden, en niet te vergeten ook deze drie (we hebben `\le@ding` gedefinieerd als `\normalbaselineskip`):

```
\bigskipamount = \le@ding
\medskipamount = .5\le@ding
\smallskipamount = .25\le@ding
```

4.4 Displays en insertions

Hier is het optimaal bereikbare ervoor te zorgen dat de display of insert een exact aantal regels hoog is, zodat verderop een en ander weer goed gaat lopen.

We hebben twee soorten macro’s nodig, namelijk één die in staat is om een `\box` af te ronden naar de juiste hoogte + diepte, en één om de bedoelde tekst in zo’n `\box` te vangen.

Het aanpassen van de hoogte gebeurt met de macro `\fixbox`. Eerst de definitie, en dan kunnen we nog even stilstaan bij de problemen die bij de implementatie te pas kwamen.

Wat we willen kunnen doen is:

```
\def\cs#1{\setbox0\vbox{#1}\fixbox0}
```

en ook

```
\def\cs#1{\setbox0\hbox{#1}\fixbox0}
```

omdat bijvoorbeeld figuren en tabulars de neiging hebben om `\hbox`-en te zijn. Het verschil blijkt weinig uit te maken. De structuur voor `\fixbox` is:

1. Meet de hoogte en diepte van de `\box`
2. Zoek uit hoeveel regels dat geweest zijn
3. Neem nu 1 regel meer als zijnde het gewenste aantal
4. gewenst aantal regels \times hoogte van 1 regel = gewenste hoogte
5. Nodige toevoeging is gewenste hoogte – echte hoogte
6. toevoegen aan de hoogte (niet de diepte!), maar wel centreren
7. zetten van de `\box`

De ‘echte’ `\fixbox` schrijft daarnaast nog het een en ander aan debugging informatie naar het log. Dat slaan we voor het gemak over.

```
\newdimen\addedheight
\def\fixbox#1{%
  \@tempdima=\ht#1
  \advance\@tempdima\dp#1
  \@tempdimb=\dp#1
```

De totale grootte staat nu dus in `\@tempdimb`

```
\divide\@tempdima 11
\@tempcnta=\@tempdima
\divide\@tempcnta 65536
```

`\@tempcnta` is het aantal gevonden regels, afgerond naar beneden. Jammer van die maffe `\divide`, maar een assignment naar een counter vanuit een dimen register geeft een getal terug in scaled points.

```
\@tempdima=11pt
\advance\@tempcnta1
\@tempdima=\@tempcnta\@tempdima
```

De gewenste hoogte staat nu in `\@tempdima`. Dat kon, want we hebben de originele hoogte niet meer nodig in het scratch register. In plaats daarvan komen die gegevens uit het origineel:

```
\addedheight\@tempdima
\advance\addedheight-\dp#1
\advance\addedheight-\ht#1
```

Ziezo. `\addedheight` bevat nu de nodige toevoeging, en na de volgende regel

```
\advance\@tempdima-\dp#1
```

Is `\@tempdima` de gewenste uiteindelijke hoogte. Nu moet de `\box` nog overnieuw verpakt worden, en de diepte hersteld, en dan zijn we klaar.

```
\setbox6=\vbox to \@tempdima{%
  \vfil\kern \addedheight\box#1\vfil}%
\dp6\@tempdimb
\box6
}
```

Niet echt ingewikkeld dus, achteraf. Maar dat neemt niet weg dat dit al minstens de vijfde versie van deze macro is, en dat er nog steeds allerlei grensgevallen zijn waarin deze oplossing niet voldoet. Vooral het laatste stukje (het overnieuw inpakken van de gebruikte box) bleek nog flink lastig.

- De allereerste versie van de macro vergat bijvoorbeeld de diepte te herstellen, met als gevolg dat de hoogte van de box een paar punten teveel werd. Dat klinkt onschuldig, maar daardoor werd het verschil tussen de box en de vorige regel (die immers zelf ook een diepte had) te klein, met als effect dat alle tekst onder een `\fixbox` een punt of twee te laag uitkwam.
- Een soortgelijk probleem deed zich voor toen ik probeerde de toegevoegde ruimte eerlijk te verdelen tussen hoogte en diepte, maar nu was de toegenomen diepte de boosdoener.
- De twee `\vfils` zijn dan weer het gevolg van het feit dat veel figuren in een box staan met diepte nul, en daardoor slecht gecentreerd uit de vorige versie van `\fixbox` kwamen.
Enzovoort.

Environments Nu dient natuurlijk het class file er ook voor te zorgen dat de standaard omgevingen die last hebben van de verschuivende baselines aangepast worden.

We zullen u verder niet vervelen met alle verschillende definities voor alle environments, maar een simpel voorbeeld is het abstract. Omdat de regelafstand van het abstract kleiner is dan voor de broodtekst, is het een ideale kandidaat voor aanpassing. Het zag er eerst zo uit:

```
\newenvironment{abstract}
{\RaggedLeft
 \small\sffamily
 \textbf{Abstract}\par\noindent
 {%
 \par\vspace*{10pt}
 }
}
```

En dat ziet er nu zo uit:

```
\newenvironment{abstract}
{\setbox\@tempboxa=\vbox
 \bgroup
 \RaggedLeft
 \small\sffamily
 \textbf{Abstract}\par\noindent
 {%
 \par\vspace*{10pt}
 }
\egroup
\fixbox\@tempboxa
}
```

5 Diversen

Eén smerige truc is nog wel het vermelden waard. Het class file gebruikt ook een ‘emulatie-mode’ (wederom met dank aan Siep), waarin weliswaar niet exact de juiste fonts gebruikt worden, maar waarin Helvetica en consorten zo worden aangepast dat de meeste regelafbrekingen redelijk kloppen (let op: dit zegt niets over de typografische kwaliteit van de uitvoer).

Onder het NFSS betekent dat normaal gesproken dat er een stuk of drie-vier extra font definitie bestanden meegestuurd moeten worden. Dat is onhandig, maar gelukkig is er een truc om daaraan te ontsnappen:

```
\begingroup
\nfss@catcodes
\Declare..... % kortom, het hele .fd bestand
% minus \endinput
\endgroup
```

6 Wat er nog te doen valt

Aan de macros: eigenlijk zou het hele `\section` mechanisme aangepast moeten worden.

Het class file geeft anderhalve witregel boven de kop en een halve regel onder de kop, en in de gewone tekst functioneert dat naar behoren. Maar als er een paginabreuk komt net voor het kopje (en dat is iets wat we eigenlijk graag willen) blijft alleen de onderste `\glue` over, ter grootte van precies een *halve* regel.

Voorlopig passen we dit aan door met de hand wat aan de witruimte te rommelen, maar het moet wel automatisch kunnen. Ik heb wel een paar ideeën, maar implementatie is verre van triviaal, dus de definitieve afloop daarvan zal op de volgende MAPS moeten wachten. In ieder geval zijn er in grote lijnen (minstens) twee oplossingen mogelijk:

- Eerst de bovenste witruimte (een glue) toevoegen, en vervolgens de rest (tekst plus onderste witruimte) in een `\fixbox` plaatsen. Deze oplossing is redelijk eenvoudig, maar heeft als nadeel dat de baseline van de tekst van de kop niet meer op een stramien-regel staat.
- Maar het is ook mogelijk om eerst de glue te doen, dan een penalty, en dan kijken of de pagina afgebroken is (dat kan door te kijken of het register `\pagetotal` gelijk aan nul is). Zo ja, dan is vervolgens vrij eenvoudig de witruimte onder het kopje aan te passen vanuit de macro zelf.
Dit lijkt een betere oplossing, maar bestaat uit veel meer werk, omdat de door \LaTeX 2_ϵ aangeboden low-level macros (`\@startsection` en dergelijke) hierbij absoluut nutteloos zijn.

PDF zaken: de ondersteuning voor `pdftex` kan nog heel wat beter, en is daarnaast feitelijk nog nauwelijks getest. En voor de versie die uiteindelijk op het web gezet gaat wor-

den is het eigenlijk wel prettig als we nog een alternatieve pagina-layout bedenken die zich beter leent voor lezen online.

De documentatie: ontbreekt tot op heden vrijwel volledig. Dat kan natuurlijk ook weer niet als we willen dat auteurs het class file gaan gebruiken.

7 Afsluiting

Wat dankwoorden zijn hier op zijn plaats. Allereerst Siep Kroonenberg natuurlijk, voor het professionele ontwerp, het aanmaken van de benodigde NFSS en TFM bestanden, en de eerste implementatie van de macros. En dan uiteraard de redactie zelf (en niet te vergeten Frans Goddijn), wegens de geboden feedback tijdens het testen en ontwikkelen. Hans Hagen voor de verbatim macros en Kluwer Academic Publishers voor de float macros. Als laatste iedereen die in het verleden aan de vorige versie gewerkt heeft, want zonder de oude was er beslist ook geen nieuwe geweest.

Het bestand `maps.cls` staat natuurlijk op de NTG web pagina (redactie-pagina) en ook op de nieuwe \LaTeX CD-ROM. Zodra de documentatie klaar is zullen het class file en de bestanden voor de font-familie Frutiger ook op CTAN gezet worden.

Bijlage 3

Book Design for T_EX Users

Part 1: Theory

Philip Taylor
The Computer Centre,
Royal Holloway and Bedford New College,
University of London,
Egham Hill, Egham, Surrey TW20 0EX,
United Kingdom.
Philip Taylor (RHBNC) <P.Taylor@Vax.Rhbnc.Ac.Uk>

abstract

Book design cannot be taught; it can only be learned, preferably by critical study of as many books as possible. Of all the elements which make up a book, white space is frequently the least considered and the most important. *Avant garde* designs are compared and contrasted with more conservative and traditional approaches. Three key elements: *uniformity*, *information* and *structure* are identified, and ‘good design practice’ discussed in terms of each of these.

keywords

Design, typography, layout

There can never be too little space below headings,
only too much!

1 Introduction

The widespread use of T_EX and other typesetting or DTP packages by tens of thousands of scientists, researchers and other academics has resulted in two rather disturbing phenomena: (1) more and more people are spending ever longer trying to get their publications to *look* right, rather than worrying about whether such publications are factually correct or are well written, and (2) fewer and fewer people, on opening a book for the first time, think first about the content, but instead commence by judging the book on its form, or to be more precise, on the appearance of the design and typesetting. We are, in fact, becoming a generation of self-taught designers and typographers, but in so doing we are tacitly avoiding the many years of training, apprenticeship and indenture which previous generations have deemed necessary.

This is, in itself, no bad thing — there are far too many self-appointed ‘experts’ ever ready to initiate neophytes into the arcane mysteries of their craft, in exchange for not

inconsiderable sums of money — but in order for learning by osmosis to be effective, the beginner has to be exposed both to good and to bad examples of the art, and to think critically about what it is that differentiates the former from the latter. In Departments of Typography and Design, such examples abound, and the professors daily compare and contrast good with bad to the great benefit of their students; but in the incestuous world of T_EX, good examples are rare whilst bad examples abound.

But why should this be? What is it about T_EX, which in skilled hands is capable of producing results equalling the very best examples of hot lead composition, that encourages the production of second- and even third-rate designs? I suggest that there are two main answers to this: (1) in *The T_EXbook*, which is presumably the first (if not the only) book on typesetting that users of T_EX¹ encounter, there is extraordinarily little guidance given concerning document design, as opposed to document formatting, and (2) the standard styles which accompany L^AT_EX generate results which even the staunchest L^AT_EX adherent would have difficulty arguing represent “the state of the art” in document design, and which if considered dispassionately might justifiably be said to lack subtlety and finesse.²

Thus the lack of explicit guidance, together with the rather poor examples generated by the standard L^AT_EX styles, has resulted in a proliferation of poorly designed books all of which shriek “T_EX” (or “L^AT_EX”).³ It would not be fair on the authors to adduce particular examples of this creed of mediocrity, but a glance at any reasonably complete library of T_EX-related (or even T_EX-set) books will shew what I mean. . .

However, all is not lost: a new generation of T_EX-setters are emerging who appear to have studied the typesetter’s craft, and several of the more recent books on T_EX shew every evidence of having been *designed*, rather than having been ripped untimely from their progenitor’s womb.

1. as opposed to L^AT_EX.

2. The Dutch, always sensitive to such issues, have produced a sub-style ‘Sober’ which attempts to tone down the worst excesses of the default L^AT_EX styles.

3. Knuth, in his closing exhortation, wrote: “GO FORTH now and create *masterpieces of the publishing art*.” Nowhere, so far as I can trace, did he write: “and let every one of them shriek ‘T_EX’ from every page” . . .

In this paper, then, I propose to discuss what it is that differentiates a well-designed book from one that is ill-designed (or worse, one that has not been designed at all); and in so doing, I hope that I can in some small way contribute to a more universal adoption of T_EX. For all the while that professional publishing houses see only bad examples of T_EX-set books, they are unlikely to consider adopting T_EX as a house standard; but if the general standard of T_EX-set books can be raised to a point where they are either indistinguishable from, or even better than, books produced by traditional means or by commercial typesetting packages, then simple economics will ensure that such publishing houses give T_EX the consideration it so richly deserves.

2 The Book

We all know what a book is, for we handle them every day; assuming a Western culture, it is basically a set of uniformly sized sheets of paper, joined at their left edge in some way, sandwiched between two slightly larger sheets of a more rigid or robust material that wraps around the left edge. It is differentiated from a magazine primarily by virtue of its cover: that of a magazine is only slightly more robust (although frequently more glossy) than the pages which it protects, whilst that of a book is almost invariably either thicker, or more rigid, or both; a magazine cover, too, has only one basic point of articulation, whilst most book covers articulate independently of each other. One other aspect separates the all but the thinnest book from all but the thickest magazine: a book is typically bound in *signatures*, whilst a magazine is usually stapled through its spine as a single entity.

But open a book and open a magazine, and we see that these differences are only superficial; for there are far more fundamental differences which manifest themselves once inside the cover. The magazine is characterised by variation — each page is clearly different from the preceding and the next; whilst the book is characterised by uniformity — each page, seen from a distance, is virtually indistinguishable from the next (special pages apart). And in this uniformity lies the basis of successful book design; for readers have come to *expect* this uniformity, and anything which detracts from it will serve only to distract the reader.

Yet uniformity of itself is not enough: we could achieve uniformity by leaving each page blank, or by simply placing a large black rectangle within the margins of each page; but this will not satisfy our reader, who looks not only for uniformity but for *information*. Indeed, information is the very *raison d'être* of a book: without it, the book serves no purpose at all, and is at best a work of art (and at worst is totally valueless).

So the book exists to supply information; and anything which inhibits or interrupts the flow of information from book to reader will diminish its value. If the flow of information is too badly affected, the reader will simply cast the book aside (how many of us, on attempting to read a page of reversed-out Bodoni in some otherwise traditional magazine, have simply given up and left the material unread? I have, many times, and cursed the designer for his/her stupidity in putting form before function).

Uniformity, information: what else? Well, if the book is in any sense *technical* (by which I exclude the novel but include almost everything else), then it is also *structured* (indeed, as we shall see, even a novel is structured in many senses, but not in the one which I am using here); and, possessing structure, it is capable of being accessed in a structured manner. It will have, at the very least, a table of contents; it *should* have an index (although far too many books that would benefit enormously from an index are lacking in this respect), and it may also possess an internal structure, in that the reader may be asked from time to time to *see Chapter 3*, or *see also Section 2.4.2*.

And these three elements, I suggest, lie at the heart of successful design: uniformity, information, and structure. We will look at each in turn to see how it may best be achieved, implemented or accomplished.

3 Uniformity

Take a book (a traditional book, not one hot off the presses of the DTP revolution), and flick the pages, rather as if there were one of those old-fashioned animated cartoon characters lurking in the corner of each page. What do you see? Most people perceive a regular grey grid: not black and white — you only see that if you look at a static page — but instead a grey blur where the text appears, and white where there is no text (or other material). What is significant is that the white appears at the same place on every page: above the headline, below the footline, between the head/footlines and the body of the text, and to left and right of the text, in the margins of the page. If the book is set in multiple columns (usually two, but rarely more, except for rather specialised works), then a further block (or blocks) of white space will appear, separating the columns from each other.

And in many senses, this white space is the most important of the graphic elements which will go to make up each page. It provides the framework or matrix within which the ‘dark matter’ — the text, graphics, etc., which make up the *information* content of the page — is set. But probably because it does not of itself appear to carry any information, it is frequently afforded less respect than it deserves, particularly by those undertaking design without any formal background. And yet, although it does not *appear* to carry

any information, in fact it carries a great deal: without it, we would not know where the headline stopped and the page body began; where the page body stopped, and the footline began; where the left column stopped and the right began, and so on... In fact, it is *vital* to our comprehension of the contents of the page, and is therefore *at least* as important as every other element on the page, if not more so.

Because the white space and the dark matter are inextricably interlinked — one starts wherever the other stops, until the physical limits of the page are reached — any discussion of the uniformity of white space must equally be interlinked with a discussion of the uniformity of the dark matter of the page. But there is a third element to this uniformity which is even more dependent upon the inextricable interlinking of white space and dark matter, and that is the sense of ‘greyness’ of every page. The human eye is remarkably sensitive to small variations in grey level, and if the apparent greyness varies either within a page or between pages (particularly between facing pages, which form a *spread*), the effect can be quite disconcerting. Such variations in apparent greyness can result from a variety of causes, of which the most common are: (a) use of letterspacing for justification; (b) inconsistent leading between two or more blocks of text in the same font; (c) inappropriate changes of leading or font (or both) when deliberately setting a block of text in a different font (in a multiline quotation, for example). The cures for each of these ills are fairly straightforward: *never* use letterspacing to achieve justification, unless its use is so subtle that the eye cannot perceive the variation in inter-letter spacing; *never* allow the typesetting system to vary the leading in order to achieve vertical justification (and never set two blocks of text in the same font but with a different leading without being aware of the effect which will be achieved); and be aware of the perceived grey-level (white-matter : dark-matter ratio) when setting blocks of text in different fonts.

In an ideal world, attention to the suggestions of the previous paragraph would do much to ensure that the apparent greyness of each page was uniform: but there is another problem which results from our less-than-perfect world which can also significantly affect perceived greyness, and this is the problem of ‘print through’. Ideal paper presents a uniform opaque whiteness on which the black of the ink is superimposed; real paper, on the other hand, whilst uniformly white (at least, as far as high-quality printing papers are concerned) is rather less than opaque; if held up to a bright light, even the best paper will allow some light to shine through, and poorer papers are so translucent that printed material can be read almost as easily from the back as from the front (albeit as a mirror image). This in itself would be no problem were it not for the fact that the two sides of each sheet are logically independent entities: not only are they printed in separate operations, but

the material appearing on one side bears little or no correlation with the material appearing on the other. However, in *designing* such pages, the effect of their back-to-back nature must be borne in mind, and a good design will attempt to ensure that each line of text on the obverse is matched by another line of text on the reverse. Of course, in practice this is not achievable; sections break up the flow of the text, as do illustrations and other graphics: but it must be the *intention* of the designer to achieve this line-for-line equivalence, and on this philosophy is predicated the whole concept of the *grid*.

The grid represents an abstract model of each page; special pages (e.g. chapter openings) may be afforded a special grid of their own, but normal ‘running’ pages will each use the same grid, onto which is mapped the various elements of the page. The grid can be perceived in a hierarchical manner: at its most superficial, it will have lines for the physical limits of the page, for the left and right limits of the text (or of the columns, if a multi-column work), for the upper and lower limits of the page body, and for the headline and footline. At the next level of refinement, the page body will be divided into lines of text (which is why most traditional specifications for books express the dimensions of the page body in terms of lines of text, rather than so many picas or so many inches or centimetres). Superimposed back-to-back, two of these grids will intermesh perfectly, each line of text on one side corresponding to a line of text on the other; the left edge of the text on the obverse will exactly align with the right edge of the text on the reverse, and so on (which has implications for the margins, as we shall see).

Of course, the grid is an ideal, but conformity to the grid must at times be allowed to be violated; if this were not the case, there would only be a finite number of positions at which a heading (for example) could appear above the text which it introduces: one line, two lines, three lines, etc. But such granularity is far too coarse for the aesthetic demands of real book design, and headings therefore need to be treated as special cases, allowed to float away from their ‘natural’ grid line whilst the paragraphs of text above and below the heading remain bound to the grid. Illustrations and graphics, too, must be treated as special cases, and float within the white space equivalent to an integral number of lines of text, thereby themselves being independent of the grid whilst leaving their surrounding paragraphs locked firmly in place.

But sometimes the requirements of page makeup will dictate that a particular page be underfull: a paragraph, for example, may finish in such a way that there is insufficient room (e.g., only a single blank line) for a new paragraph to start; or there may be room for a heading but not for a heading plus post-heading vertical white space plus at least two lines of text. How, in those circumstances, can the con-

trading requirements of uniformity and page makeup be reconciled? In the limiting case, there is no general solution which will always work, and practical (real-life) books may occasionally have to violate one or other constraint; but equally often there is a solution which is both elegant and aesthetically pleasing: violate, by the same amount, the constraint of uniformity for two facing pages (i.e., for a *spread*). If, for example, the verso (left-hand) page runs one line short, then *force* the recto (right-hand) to run one line short also; if the verso page would ideally run one line long, then allow it so to do, but require the matching recto page to run one line long also.

And in this concept of balancing the *spread*, as opposed to achieving uniformity between *all* pages, lies, I believe the essence of good design. For when the book is held open in the hand, or laid open on the desk or lectern, it is not a single page that is seen at all, but a double-page spread; and if the two facing pages of the spread appear uniform (uniform in greyness or visual density; uniform in placement of headline and footline; uniform in size of margins — outer margins the same size as each other, inner margins also the same size but not necessarily the same apparent⁴ size as the outer; and uniform in terms of grid-lock, in that verso lines appear in perfect vertical alignment with their recto counterparts) and balanced (with both verso and recto page bodies starting at the same height from the bottom of the page, and extending for the same depth), then much will have been accomplished; and if this same uniformity and balance can be carried through every spread of the book (thereby avoiding problems of print-through and so on), then much of the framework of good design will already be in place.

But there are many practical problems associated with the concepts of gridlock and balanced spreads; some of these are particularly true when using T_EX, whilst others are more general. Those that are particular to T_EX will be addressed in the sequel to this paper⁵, whilst those that are more general are discussed below.

Considering first the problems of balanced spreads: it was suggested above that if the natural height for a verso page was one line short, or one line long, then it should be set to its natural size and its counterpart recto page *forced* to the same size. But what if the verso page naturally sets at the target size of the page, whilst the recto page runs one line light or one line over? If page makeup is performed on a page-by-page basis, then it is already too late to re-set the verso page, and either the recto page will have to be set to a non-natural size (if there is sufficient flexibility in the page makeup to allow this), or the balance constraint for the spread violated. And therefore we must postulate that any typesetting system intended for the production of well-designed books *must* be capable, at the very least, of setting a *spread* as an entity, rather than a page. Of course, this

does not let us off the hook completely: for example, if the verso page naturally runs one line over, but the recto page finishes a paragraph at the natural height for the page, then it may not be possible to graft an additional line onto the recto page without violating some other (tacit) constraint; in these circumstances it may be necessary to backtrack even further, and to start asking questions such as “what if I were to set the preceding spread one line light, or one line over”, and so on; in the final analysis, the more decisions about page makeup that can be deferred, the better the final volume is likely to be. As computer memory becomes cheaper and cheaper, it is by no means unreasonable to think about optimising a complete chapter at a time.

And what of uniformity: what if a multiline quotation, set in a smaller font with correspondingly reduced leading, *must* appear as an entity on a page, whilst there is no matching quotation on the other (physical) side of the same page? Then print-through will undoubtedly occur for the duration of the quotation, and in the worst case there will be an interference effect as the lines of the quotation drift into and out of synchronism with the lines of paragraph text on the other side. Here no matter how much material we accumulate can a makeup solution be postulated; and in the end we are dependent more on the skills of the paper maker in achieving near opacity than we are on our own skills in using and programming our typesetting system.

But there is much more to uniformity than simply gridlock and balanced spreads. Uniformity is a concept which percolates every element of good book design. Consider, for example, the treatment of chapter headings, section headings, paragraphs, quotations and so on: in what sense can they, too, be made ‘uniform’? Clearly each must be unique, in order for the reader to immediately identify at what sort of entity he or she is looking; yet if they are not only unique but are also afforded wildly disparate typographic treatment, then any sense of coherence is lost and the book starts to take on the appearance of a mismatched hotchpotch of design ideas.

We might start by positing that there should only be a small number of different fonts used — ‘the fewer the better’ is hard to equal as an axiom for the selection of fonts! — whilst equally there should only be a small number of placements. For example, if paragraphs are fully justified (as would usually be the case for a book, although exceptions to this rule will be discussed elsewhere) and if section headings are set ranged left, then the book as a whole should probably restrict itself to these two styles of setting:

4. I use the term *apparent* here quite intentionally, for as we shall see, the *apparent size* of the inner margins is always less than their actual size, by an amount which is a function both of the thickness of the book and of the binding technique(s) used.

5. Book Design for T_EX Users; Part 2: Practice, elsewhere in this volume.

it would normally be inappropriate to have centered headings in a book that otherwise has a fully-justified or ranged-left theme running through it. But if section headings are set ranged left (perhaps in conjunction with semantic line breaks⁶), whilst normal paragraphs are set fully justified, then quotations could either be set fully justified (like paragraphs) or ranged left (like section headings), but should probably not be set ranged right without good reason.

And what of indentation? Here two different schools of thought obtain. One would argue that the requirement of uniformity encompasses indentation, and that the indentation, once chosen, should apply to the whole book: thus, for example, lists would be indented by the same amount as paragraphs; quotations might be set with an additional left margin equal to this indentation; and the bibliography might be set with reverse indentation also equal to this same amount. The other would say that the requirements of clarity and lack of ambiguity dictate that a *different* indentation should be used wherever different entities occur, thereby giving the reader maximum indication of the nature of the entity being indented even on the most superficial glance at the page. I have sympathy with both points of view, but my inherently conservative background renders the former more appealing than the latter; I do not think I have yet seen an example in which the reader could have been misled had a uniform indentation been adopted. But this whole area transcends the boundary between *uniformity* (which suggests a uniform indentation), and *information* (which suggests different indentations for different purposes), and brings us naturally to the next section.

4 Information

The primary function of any book is to convey information; yet the preceding discussion has concentrated almost entirely on the aesthetics of book design, rather than on its rôle as a medium for the communication of information. However, provided that the two ideas do not come into conflict, a uniform and aesthetically pleasing appearance does much to assist the book in its communication rôle, for it allows the reader to concentrate on the text (i.e., the *information content* of the book) whilst not being distracted by its design (a fact which is sadly ignored by many of today's more *avant garde* designers). But there comes a point at which excessive adherence to the precept of uniformity would start to detract from the book's primary rôle as information source, and it is therefore to this area that we must now turn our attention.

Consider first of all the rôle of section headers: those single (or occasionally multiple) lines of text which serve to introduce the reader to the ideas which follow. This paper, for example, makes use of only a single level of section header, the author preferring to lapse into straight prose

within each section; other authors, particularly those with a strong scientific background, feel happier if they can classify their ideas in a strongly hierarchical manner, and frequently have recourse not only to *A-heads* (as in this paper), but B-heads, C-heads, D-heads and even E-heads on rare occasions. The first requirement for such headers is that they shall, *unambiguously*, refer to the text which follows: it should not be possible, in a well-designed book, to mentally attach them to the preceding text. The means by which this is accomplished is simplicity itself, yet is so often violated in amateur-designed books and other documents that one wonders whether the idea has ever occurred to their designers at all: the section header shall be physically closer to the text which it introduces than to the text which precedes it. Note that this is strictly a 'less than' relationship, not a 'less than or equals' one: the header must *never* be equi-spaced between the preceding and following texts. This rule has some interesting knock-on effects: for example, a header must *never* appear in isolation at the bottom of a page, for were it so to do, it would by definition be nearer to the preceding text than from the text which follows.

But in a strongly hierarchical book or paper, it is just as important that the different levels of header (A-head, B-head, etc.) shall be capable of being differentiated at a glance. How should this hierarchy of headers best be conveyed to the reader? We have available several options: (1) Higher-level headers may be separated from their preceding text by greater amounts of vertical white space than lower-level; (2) Higher-level headers may be separated from their qualifying (following) text by greater amounts of vertical white space than lower order; (3) Higher-level headers may be set in a larger font than lower-level; (4) Higher-level headers may be set in a bolder font than lower-level; (5) Some other typographic differentiation (e.g. the use of a *sans serif* font in a book or document otherwise set in a *serif* font) may be used for one or more levels of header; (6) Run-in headers may be used for the lowest level of header. Indeed, these are only some of the available options: for example, in some works a new page is taken for each new top-level section, even where that section is only one of many similar sections in a chapter.

Clearly the range of options is vast, and it is not possible in a paper of this brevity to give more than a few typical conventions, but one requirement is tantamount: if two or more conventions are adopted within a single document, then no combination of those conventions must lead to ambiguity. For example, if A-heads are set in 16 point roman, B-heads must not be set in 14 point bold, for the boldness

6. A concept whereby a ragged-right setting is used in conjunction with 'strongly recommended' line breaks, thereby ensuring that complete ideas (phrases, clauses, sentences, etc.) are not unnecessarily split over two lines.

of the B-head would counteract the effect of the smaller font and lead to ambiguity in the mind of the reader. Even if a bold font is not explicitly used, it is possible (by, for example, selecting an ill-matched *sans serif* font for B-level headers in an otherwise *serif* document) to accidentally specify an *apparently* bolder font for a subsidiary-level header. Such ambiguities must be avoided.

In what other ways can the book designer ensure that information is most clearly conveyed? Perhaps most important of all by ensuring that the book can be *read!* This goes without saying, you may say, but there are sadly only too many counter-examples already published for this particular requirement to be omitted from any reasonably critical analysis. Perhaps we need to start by defining what we mean by “to read”; I suggest that if reading is to be conducted efficiently and pleasurably, then it must (for the normally-abled adult) be an almost unconscious activity. If I pick up a book hoping to gain information from it, then the *last* thing that I want is to have the designer’s personality forced down my throat (unless it is a book on book design, in which case I may be able to judge from the book’s design whether or not to bother to read it!); the design must therefore be very ‘quiet’ and unintrusive, allowing the content to flow naturally forth through the medium of the form, rather than having the form leap out from the page and distract the reader from the content. Naturally there are exceptions to this rule, and books on design clearly come into that category, being inherently self-referential, but generally speaking the reader wants to know as little about the designer and as much about the content as possible.

Furthermore, reading must be able to proceed in a linear and uninterrupted manner; it is well known that any infelicity on the part of the author which results in ambiguity in the reader’s mind will cause the latter to back-track through the work, hoping to gain further clues and thereby disambiguate the text on a second or subsequent reading. Classic authors on grammar (Fowler, Weseen, Partridge, Onions, Gowers, Quiller-Couch, Sweet) pay much attention to this. But there are many typographic pitfalls which can also cause a reader to have to backtrack, and it is as important for the designer to avoid these as it is for the author to avoid the grammatical infelicities.

For example, during the 1930s, there was a great vogue for *sans serif* faces: they were modern, *avant garde*, stylish, modish — use whatever term you will. And particularly in North America, and to a lesser extent in Europe, such was the pressure to use these typefaces that their *raison d’être* — to provide a simple, minimalist, style for short sections of text which would not draw attention away from the main theme (frequently an accompanying graphic) — were forgotten, and they were advocated (and used) as *the* typefaces for every conceivable purpose.

These purposes were not restricted to their classic use in headings, captions, posters, etc., but were instead extended to encompass even the running text of books; every page was set in *sans serif* text, with little feeling for the comfort and convenience of the reader. The effect on the reader was all too predictable (with hindsight): readers found it difficult to concentrate on such books for any period of time, finding it tiring and even distressing; and the reason was very simple, although not well understood at the time: even though the *serifs* which characterise most of our classic typefaces today are in reality no more than artifacts dating back to the original letterforms of stonecutters (particularly in the case of upper-case letterforms), and later of typecutters, these *serifs* perform a very important function when the letterform occurs in running text: they serve to draw the eye naturally along the line of text, very much reducing the risk of the eye vacillating between two adjacent lines of text, and also help to minimise the amount of backtracking within a single line. And so, with the benefit of hindsight and of psychological and physiological research, it has now been established that the typeface of choice for passages of running text (as opposed to captions, etc, which extend for at most a few lines) is almost invariably a *serif* face. Sadly this fact is still occasionally ignored.

But if the choice of a *serif* face is almost mandatory to avoid vacillation between lines of text and backtracking within a single line, what other psychological or physiological factors can also affect the readability of the text? Perhaps the most important of all, and one for which plain \TeX sadly gives most inappropriate guidance, is the size of font with respect to the *measure* (i.e., the width) of the text. Plain \TeX is predicated on the use of 10 point fonts on a measure of 6.5 inches (39 picas), which simply gives far too many characters per line. Psychologists have shown that the optimal number of characters per line for normally sighted people lies in the range 40–70, and peaks somewhere near the upper bound of that range; below it, people become frustrated: they are forced to take in too little information per glance; and above it, they tend to lose their place, and either backtrack within the line, or on re-scanning to the start of the next line, lose their vertical place and re-scan to the start of the wrong line. Even \LaTeX , which generally gives better guidance than plain \TeX in matters of typographic design, allows the user complete freedom to select between 10 point, 11 point and 12 point fonts, regardless of the style chosen and therefore of the measure of the text. For Europeans readers, accustomed to the DIN series of paper sizes, the best guidance I can give is as follows: if you are setting on a sheet of A4 paper (which is unlikely for a book but quite possible for a report or other similar document), with ‘normal’ margins (circa 1 inch), then a 12 point font is called for; you can get away with 11 point,

but 10 point is out of the question. The same goes for North American readers with 1 inch margins on a sheet of American ‘letter’ paper, 8.5" × 11". And for a book? Well, ‘how big is a book’ is a question to which I will return in the sequel to this paper, but generally speaking books *are* set in 10 point typefaces; however, as the width of the paper increases, two columns become obligatory or pathologically large margins become required.⁷ In unusually small books, 9 point fonts may be used, but anything less than this poses problems of legibility for normally sighted people.

In the preceding paragraph, I have spoken of a “10 point font” as if it were some sort of ISO standard; but sadly it is anything but. Fonts vary enormously both in their actual size (as measured), and in their perceived size, and the quoted size is at best an approximation and at worst a d@mned lie! For what it is worth, the notional size of a font is that distance which may separate consecutive lines of text in a paragraph set in that font without the descenders of one line overlapping the ascenders of the line below; it is also approximately the height + depth of a parenthesis glyph. But in practice one designer’s 10 point font may well be another’s 11 point; and if you are using two or more fonts in a single document, then it is your responsibility as designer to ensure that the size at which they are used renders them visually conformable, even if this means loading one at 10 point and another at 11 point (or even at 10.6347 point, if that represents the true ratio between their perceived sizes).

And for the leading: some authorities will suggest “1.2 times the design size of the font”; others will suggest “2 points more than the design size of the font”; and others will suggest yet further formulæ. The answer is, of course, that no one formula will be right for every font, or for every size, and until experience has given you the insight to look at a font sample and *know* the appropriate leading for the target font size, then you will have to use the most powerful tool available to you: your eyes. In other words, you will have to print samples of the text at various leadings (probably of the order of magnitude suggested by the formulæ above), and adjust until it looks right to you. But when you print these samples, you will come up against another, and very subtle, psychological quirk: assume you do as most people do, and print your proofs on a laser printer; then your output will appear either on a sheet of A4, or on a sheet of ‘letter’ size paper, and most unusually on anything else. And try as you might, you will not be able to judge the size of the font and the size of the leading as they will appear in the final book form, even if you draw a box around your sample text to represent the dimensions of the final trimmed page; your eye/mind will refuse to believe that the white paper which lies outside that line is not attached to the text, and will judge the size of the text and the

size of the leading in terms of the untrimmed sheet of A4 or ‘letter’ paper. The solution, of course, is to guillotine the paper to the final trimmed size, and then to paste two such trimmed sheets together (or to print a double page spread in the first place) and to look at a full-size replica of the final spread of the book; and then, and only then, will you be able properly to judge the size of type and the size of the leading in terms of the printed page.

5 Structure

Finally we turn our attention to *structure*, and in particular to the means by which a well-designed book can be efficiently referenced (and cross-referenced) in a quasi-random, rather than sequential, manner. At the coarsest level of granularity, a book is divided into volumes (if huge), parts (if large) and chapters (almost all books). Access to volumes need not worry us unduly: each will contain the name and/or number of the volume on the spine and front cover, and only if two or more volumes are concurrently open in front of the reader will it be necessary to be able to differentiate between volumes by inspection of only the open spreads.

Parts are not uncommon, but many of the potential problems associated with the identification of parts can be eliminated by sequential numbering of chapters independent of the part in which they happen to fall; with sequential chapter numbering, the reader can always be referred to *Chapter n*, without needing to qualify it as *Chapter n of Part m*.

But the most important division of the majority of books is into chapters, and here we must start our investigations into *structure* in earnest. Consider the classic case of a multi-chapter, single-volume, book, with a table of contents (‘TOC’) among the *front matter* (a.k.a. ‘the prelims’). The reader wishing to access the book through the TOC consults the latter and sees, for each chapter, its number, its name (if the chapters are named), and the page on which it commences. Selecting a chapter from those listed, the reader flicks through the pages looking for the page on which the chapter starts. This is not a random search: the page numbers increase monotonically with period 1, and if the reader overshoots he or she is invariably sufficiently familiar with the general concept of a book to realise that it is necessary to backtrack.⁸ But an interesting

7. I am advised by a North American student that it is the practice in North America for students to annotate their books; for this reason, they *expect* far wider margins than European readers, which may explain something about the default L^AT_EX styles.

8. It is interesting to realise that the scenario outlined is the converse of what usually happens in practice: because books are generally either laid on the desk/lectern or held in the right hand with the highest number page at the bottom, it is far more natural for the reader to make a *backwards* search through the pages until the desired page is found,

phenomenon occurs as the reader converges on the page of interest, at least in many less-than-optimal books: the page numbers (*folios*, as they are frequently termed) traditionally alternate between top-left and top-right, occupying the top-left placement on verso pages and top-right on recto; this placement is believed to make them maximally visible. But on opening chapter pages it is traditional to suppress the running head ('headline'), because the design of these pages (discussed in greater detail in the sequel to this paper) is such that a running head is generally considered aesthetically displeasing. And therefore the very page which (logically) bears the number sought is also the very page which (physically) has no page number on it; and the reader is forced to perform a narrow binary search to ensure that the page of interest has truly been located, by comparing the last physical page number which can be found (and which will, in the worst case, not even be visible from the page of interest, if the previous chapter happens to finish recto, since it is also traditional to start new chapters recto and a completely blank page will therefore form the verso half of the spread) and the next physical page number, which will invariably also be invisible from the page of interest. Of course, the name and/or number of the chapter will be visible on the sought page, and it will be clear from its design that it *is* an opening chapter page, but none the less the reader who until then has been searching for a specific page number is forced to modify his/her search algorithm.

The solution generally advocated for this problem is to present the page number on opening chapter pages as a *drop(ped) folio*: a centered page number occupying a part of the footline. The percipient reader soon becomes familiar with this convention, and modifies his/her gaze to take in the bottom of the page rather than the top outside edge when reaching an opening chapter page. But if dropped folios are acceptable on opening chapter pages, why not use them consistently throughout the book? This would have two beneficial effects: (1) the reader would be able to find *any* page in the book by studying the same part of every page, regardless of the nature of that page, and (2) additional space would be released in the running heads for additional (cross-)referencing material, space which as we shall see becomes of a premium as the complexity (in terms of explicit hierarchical structure) increases.

Once we have ensured that page numbers occur on *every* page (blank pages excepted, since by definition no possible well-formed (cross-)reference could require the reader to turn to such a page), we have at a stroke ensured that our tables of contents, indexes, etc., all of which generally yield a *page number* when 'dereferenced' (consulted), will invariably result in a hit rather than a miss. We must now turn our attention to other techniques for (cross-)referencing, and in particular methods for locating logical

sub-divisions of the book (e.g. sections, sub-sections, etc.) by their *name*, and also by their *number* if such entities are numbered.

Generally speaking, the names and numbers of logical sub-divisions are used for cross-referencing (i.e., referencing from within the text), rather than for direct referencing (e.g. from a table of contents or an index); but regardless of the source of the reference, the reader will ultimately be required either to *see Section 2.1.4* or to *see also Lagopus hyperboreus* — in neither case will the reader explicitly be instructed to turn to a specific page. It is frequently possible to convert one of these *indirect* references into a *direct* reference to a page number, by consulting the appropriate table of contents or index, but this two-stage process is both frustrating and time-wasting: a more direct method is required.

The mechanism by which this direct access to named or numbered logical sub-divisions of a text is generally accomplished is through the medium of *running heads*; these have been referred to previously in the current paper without any formal definition being given of their nature or purpose. A running head is so called because it recurs on (almost) every page; opening chapter pages and blank pages are usually excluded from the set of pages on which a running head can occur, and if an entire page is given over to an illustration then that page too may be excluded; but special cases apart, running heads occur on every page. But of course the *content* of the running head varies from page to page: were it not so, there would be no purpose to the running head at all (which is also frequently the case when it is used to echo the title of the book on every page or every second page; the reader is normally aware of the title of the current work, although there are counter-examples, as when consulting many works at once; thus the echoing of the title is not necessarily evidence of poor design). In general, the content of the running head is adjusted to reflect the content of the page over which it appears; thus, for example, if *Section 2.1.4: metalinguistic notions* commenced on page 23, the running head of page 23 would almost certainly reflect that fact. But in a hierarchically structured work, there are potential conflicts; consider a book with chapters, sections and sub-sections: which of these entities should the running head reflect? A convention frequently adopted is to ascribe different semantics to the verso and recto heads: the verso carries 'more significant' information (e.g. the name/number of the chapter), whilst the recto head carries 'less significant' information (e.g. the section name/number). Yet this is not enough: where should the

or until overshoot occurs, than it is to make a *forwards* search. This is because it is far easier to raise a number of pages, frequently almost the entire page set, in one hand and allow them to fall back individually under the effect of gravity than it is to lift each page individually whilst seeking the page of interest.

sub-section information appear? Ultimately there is no solution to this problem: if the book is sufficiently complex (i.e., possesses too deep a nesting), then no matter how complex an arrangement of headers is adopted there will be a level of nesting beyond which it is simply not possible to reflect lower-order entities in the header. The designer, then, must perform a trade-off, and decide which information is most beneficial to the reader. Omissions are possible at either or both ends of the spectrum: it may be that knowledge of the name of the current chapter is less important than knowledge of the current section/sub-section/sub-sub-section/sub-sub-sub-section; or it may be that knowledge of the chapter is deemed more important than knowledge of the current sub-sub-etc. The designer and author must work together on this problem.

But there is one additional mechanism which is considerably under-used, yet which allows twice as much information to be packed into each header: if folios are removed to the footline, thereby releasing the outer edge of each running head for other usage, then provided that the author can be encouraged to provide *short* names for each of his/her chapters/sections/etc., each running head can serve double duty. For example, verso heads can carry (left) chapter name/number, whilst carrying (right) section name/number; recto heads can then carry (left) sub-section and (right) sub-sub-section. Adequate space must clearly be left between the two elements to avoid potential ambiguity.

Finally, is it the *name* or the *number* of each logical entity which is to appear in the header? Above I have hedged my bets by consistently referring to name/number, yet at some point a decision must be made. If space allows, and if the author co-operates by providing short names, then there is no reason why *both* should not appear; with less space, or longer names, it may be necessary to omit the numbers in order to allow the names to appear; and if the author is unconscionably prolix in naming the various entities, then the designer may have little choice but to simply give the hierarchical name (e.g. *Chapter*, *Section*) followed by the relevant number. But this last serves the author rather than the reader, and pressure should be brought to bear on the author to provide suitable ‘short forms’ purely for use in the running heads. Of course, some works use *only* numbered entities; in such works, there is no choice: the hierarchical names (if appropriate) and numbers must be used.

6 Conclusions

Good book design can be discussed in terms of three parameters: *uniformity*, *information* and *structure* (although there are many other parameters which would be addressed in a longer paper), and attention to each of these will do much to increase the potential value of a book to its readers. More practical advice is given in the sequel to this paper: “Book Design for T_EX Users; Part 2: Practice”, elsewhere in this volume.

Bijlage 4

Book Design for T_EX Users

Part 2: Practice

Philip Taylor
The Computer Centre,
Royal Holloway and Bedford New College,
University of London,
Egham Hill, Egham, Surrey TW20 0EX,
United Kingdom.
Philip Taylor (RHBNC) <P.Taylor@Vax.Rhbc.Ac.Uk>

abstract

In the predecessor to this paper¹, three fundamental concepts of *uniformity*, *information* and *structure* were introduced, and general guidance given on each of them. In this paper, more practical advice is given, specifically in two areas: guidance on actual dimensions, proportions and layout; and guidance on implementing some of the ideas through the medium of the T_EX language. Finally, some difficult (and even insoluble) problems in layout are discussed.

keywords

Design, typography, layout

Every book will have one figure that cannot be seen from its point of reference.

1 How big is a book?

Just as we are all familiar with the general concept of a book, we are also familiar with practical upper- and lower-bounds on its size; a book that measures 3 centimetres by 2 centimetres is of as little use to most of us as a book measuring 3 metres by 2 metres. Looking at my bookshelves as I write, and ignoring only those volumes whose dimensions lie beyond the 3σ points of the distribution, I can safely suggest that the majority of ‘normal’ books lie in the range 18 cm × 10 cm to 35 cm × 25 cm. In terms of more traditional printers’ units (picas), we can re-express this range as 42 pc × 24 pc to 80 pc × 64 pc (in all cases I have approximated rather than taking any exact measurements). What is more interesting, however, is the aspect ratio of each these books: almost without exception they are in portrait orientation rather than landscape. Why should this be?

There are, I suggest, two answers to this; one intensely practical, the other slightly theoretical. The practical answer is easily demonstrated: take any book that is *not* in portrait orientation (i.e. one that is in landscape

orientation), hold it in one hand and attempt to open it: if the book is small, or tends to square rather than being overtly landscape, it will be reasonably stable in the hand, but if it is large, or markedly landscape in aspect ratio, it will tend to fold back on itself as the centres of gravity of the two halves fall outside the span of the opened hand. For certain classes of book (i.e. those intended to be read from a desk or lectern, or perhaps opened on the reader’s lap), this is of little consequence; but for those books which are most likely to be read whilst being held in the hand (which includes the vast majority of books published), such instability would render them almost unreadable, and therefore such combinations of size and aspect ratio are generally avoided.

The theoretical reason harkens back to material covered in the predecessor to this paper, and is concerned with the optimal length of line. In that paper it was suggested that between 40 and 70 characters per line is the target, with the ideal somewhere near the upper end of that range. Given that most normally sighted people can read without difficulty 9 point to 12 point typefaces at the normal distance associated with reading books, but find anything much smaller somewhat difficult to read (and tend to regard anything much larger as ‘insulting’, in the sense that it appears to have been intended for children), this suggests that most books will tend to have a *measure* somewhere in the approximate range 12 picas to 30 picas, but will tend to cluster nearer the upper end of that range. When we compare this with the range of book sizes cited above, these figures seem reasonable; the smallest book encountered was 24 pc in width, measured across the cover, whilst the largest was 64 pc, similarly measured. Allowing for trimmed pages fitting comfortably inside the cover, and ‘sensible’ margins (as yet to be defined), we find that the smallest book has a measure of 17 pc whilst the largest has a measure of 48 pc (and is set in an abnormally large font; it would be more usual to find a book of this size set in double-column format). Clearly there is a reasonable correspondence between theory and practice.

In practice, some sizes are more ‘desirable’ than others; traditionally, books were printed in a restricted range of sizes, and some of the terms used are still extant today; examples include ‘quarto’, ‘folio’, etc. Others, for example

1. Book Design for T_EX Users; Part 1: Theory

‘elephant’ and ‘royal’ have fallen into disuse, and there is today far greater freedom in choosing the final size of a book. However, practical realities intrude here, as everywhere else, and ultimately the printer will have to produce the pages of the book by sub-dividing a much larger sheet of paper; as such large sheets of paper are produced in a fixed range of sizes, it is obvious that some final page sizes will result in much less wastage than others, and such sizes are therefore to be preferred; your printer will give you advice on ‘ideal’ page sizes if asked, and will almost certainly tell you if your preferred size leads to gross wastage.

In determining the dimensions of a book, there are essentially three variables: the overall area of the text, including headers and footers; the margins; and the trimmed dimensions of the final page. Clearly at most two of these can be arbitrarily determined, and the third must follow by the simple rules of arithmetic and geometry. In practice one tends (if given total freedom) to determine the final page size and the text area first, and then to calculate the margins based on the difference; but in so doing it is important to remember that the margins are just as important as every other element of the made-up page, and cannot simply have arbitrary size. ‘Sufficient, but not too much’ is an excellent axiom to bear in mind when determining the size of margins; for example, a small book whose trimmed width is 23 pc might have an outer margin of 3 pc and a measure of 17 pc; the actual inner margin will therefore also be 3 pc, but the *perceived* inner margin will be somewhat less, as some portion of it is taken up by the binding. In general, the thicker the book the greater the apparent loss of inner margin, but binding technique is even more significant, and a well bound thick book may lose less space on the inner margin than a poorly bound thin book.

As the overall dimensions of the book increase, so may the margins; but they do not increase in direct proportion to the increase in page size: rather, if anything, they increase quite slowly, perhaps in proportion to the square root of the increase in page size, or to its logarithm. Once again, ‘sufficient but not too much’ is the key.

So far we have concentrated on the inner and outer margins, and it is worth pointing out before considering the top and bottom margins that, if symmetric perceived margins are required, this inherently requires asymmetric actual margins; but the asymmetry alternates between verso and recto pages. That is, in order to allow for the binding loss, the right margin on the verso page and the left margin on the recto page must each be increased by the binding loss. This is achieved automatically in the ‘book’ style of L^AT_EX, but plain T_EX users will need a modified output routine. In order not to need any knowledge of the existing output routine, the following code hooks into the `\shipout` primitive, and can therefore be used in conjunction with *any* output routine, no matter how complex, un-

less it, too, adjusts `\hoffset` on the fly (in which case more sophisticated code would be required).

```
\newdimen \rectohoffset
\newdimen \versohoffset

\def \bindingloss {2 pc}
  %% adjust to suit actual book
\let \Shipout = \shipout
  %% need an alias so as to steal primitive
\let \then = \relax
  %% just syntactic sugar (sorry, Kees!)

\rectohoffset = \hoffset
  \advance \rectohoffset by \bindingloss
\versohoffset = \hoffset
  \advance \versohoffset by -\bindingloss

\def \shipout
  {\ifodd \count 0
    %% can't use \pageno in LaTeX
    \then
    \hoffset = \rectohoffset
  \else
    \hoffset = \versohoffset
  \fi
  \Shipout
}
```

Before considering actual dimensions for the vertical margins, it is worth considering the simpler question of proportion, and here, as in many elements of book design, two schools of thought obtain: the first would advocate that the top margin should be less than the bottom, the second just the converse! The argument in each case is based on visual balance: those who would place the text block asymmetrically towards the top of the page claim that, visually speaking, it ‘sinks down under its own weight’, whilst the alternative school claim that unless it is set asymmetrically towards the bottom of the page, it makes the page look top-heavy and therefore unstable. My own belief is that once the effects of head- and footlines are considered, the two schools can to a certain extent be reconciled; if, however, there are no head- and footlines, then my sympathies incline more towards the ‘lower-is-better’ school than towards its opponents.

The reason for considering the head- and footlines whilst discussing the margins is that whereas the left and right margins are what I will term ‘simple’ (that is, they each occupy a single band of white space), the top and bottom margins are effectively composite: there is white space above the headline, white space below the headline, and similarly white space above and below the footline (if present; if not, then the bottom margin is simple). But in

terms of visual density, the footline is usually very light – frequently no more than an unornamented page number – whilst the headline is frequently quite dense (see the predecessor to this paper for a fuller discussion on the possible contents of a headline). The effect of this is that the two lower margins are perceived by the eye/mind as being a single band of white space, whilst the two upper margins are perceived as separate entities. The eye/mind therefore takes the sum of the two bottom margins as representing the white space at the bottom of the page, whilst more or less ignoring the lower of the two upper margins and seeing only the upper component as representing white space.

We must now attempt to summarise the preceding discussion and to come up with some firm recommendations. In general the space above the headline is significantly greater than the space below, and is of the same order of magnitude as the mean of the left and right margins (assuming for the moment that these are not exaggerated; discussion on exaggerated margins occurs later in this section). The space below the headline is fairly small: perhaps 1 pica or thereabouts. At the bottom of the page, the situation is reversed: there is relatively little space above the footline, but rather more space below. But here caution must prevail: if we were to leave the same space above the footline as below the headline (e.g., 1 pica), we would overconstrain the page makeup process, for although any page could still run one line light, it could not run one line over without interfering with the footline (or, worse, displacing the footline vertically downwards); it is therefore necessary to leave additional white space above the footline on a normally made-up page, so that an overrun of a single line can be permitted *in extremis*. Thus a gap above the footline of perhaps 2 picas is appropriate, with an additional margin of 3 or 4 picas below. Bear in mind that these figures represent only a first-order approximation, but that only relatively small adjustments would be needed for a fairly wide variation in page size.

All the discussion on margins up to this point has reflected a fairly traditional, orthodox and conservative perspective. But the size and symmetry of margins is one of those areas in which *avant garde* designers feel obliged to express their individuality. Until the advent of the so-called ‘DTP revolution’, most books had conservative margins of the order of magnitude suggested above; but at about the time when DTP was becoming widespread, a new generation of designers suddenly found the need to adopt quite enormous margins, sometimes out of all proportion to the other material on the page.² The reasons for this sudden interest in wide margins are probably quite interesting, but I suspect not well understood. I can think of several possible reasons: (1) Each generation of designers feels obliged to express its creativity in some overt manner; simply to follow the guidance of its predecessors is felt at best to be

pastiche, and at worst plagiarism. (2) The liberating effect of what I will term ‘Design through DTP’³ allowed designers to experiment with designs that might previously have been consigned to the dustbin, either because the wasteful nature of their extremes became only too apparent as real paper models were made of the design, or because the time which elapsed between the creation of a design and its first physical realisation allowed the designer time for retrospection; many, I am sure, toned down their own excesses during this cooling-off period. (3) Many of the realisations of these designs were accomplished using early DTP systems, which were themselves fairly limited in their page makeup ability; having large margins into which oversized elements could flow allowed the designers additional flexibility to work within the constraints of the DTP system.

But there is a fourth consideration, quite independent of the DTP revolution, which may also dictate the use of large margins, and this final discussion on margins concentrates solely on the page makeup problems associated therewith. Text, tables, graphics, equations and formulæ all have different, and sometimes conflicting, requirements — text, as we have seen, will normally fit best into a measure somewhere in the range 12 pc to 30 pc; tables possessing multiple columns may well not fit into such a restricted measure, a problem that also can affect complex graphics (which although generally scalable can become illegible if over-reduced); equations and formulæ may also require a measure well in excess of 30 pc if they are not to be split over more than one line. With the exception of equations and formulæ, the problems are not insoluble, or even difficult: where it is known in advance that a measure well in excess of 30 pc will be required, the text can be set in two columns whilst overwidth tables and graphics can be allowed to span both columns; as tables and graphics are generally regarded as ‘floating’ entities (that is, they can migrate in the text without causing the reader difficulty, as reference to them is almost invariable by name or by number rather than by implicit physical association), they can appear on a page in their own right, or at the top or bottom of the page on which they are referenced, without interrupting the flow of the text. But equations and formulæ (and similar entities, such as program fragments and algorithms) frequently *cannot* be allowed to float: the author will almost invariably write the text on the assump-

2. It is a sad reflection of our times that this also occurred during a period when awareness of the ecological effects of the loss of the world’s forests was becoming increasingly widespread; thus on the one hand we had the environmentalists urging us to save trees, whilst on the other we had a generation of designers apparently hell-bent on destroying the world’s forests purely to provide large asymmetric white borders for their books. . .

3. by which I mean the use of an Apple Macintosh or similar system to produce an on-screen mock-up of a proposed design without any need for a physical realisation to become available.

tion that the equation/formula will always occur exactly where it does in the manuscript, and will simply allow his or her text to 'fall through' to the equation or formula; if such an equation/formula is overlong and cannot be wrapped, then both columns of the two-column text will need to be interrupted, to the great inconvenience of the reader, for it will not necessarily be at all apparent whether the text is to be read up to the equation/formula and then continued below in the same column, or the text is to be read up to the point of the equation/formula and then continued from the top of the next column. Worse, if the equation/formula occurs not in the first column of text but the second, as the reader progresses down the first column he/she is suddenly stopped dead in his/her tracks by a completely irrelevant equation/formula; not only does the reader now not know from where to continue, he/she also does not know why the interruption occurred in the first place. Only on reading down the second column does the reason for the interruption become clear.

Therefore, in such works, an alternative approach is required, and one such approach is the use of oversized margins: the text is set to a fairly wide single-column measure, but the trim dimensions of the page are such as to allow the longest equation or formula to extend out into the (usually right) margin as necessary. The designer is then faced with another problem: how to justify to the reader the presence of these margins on pages where no such equations or formulæ occur. It is by no means unusual to find section heads pushed out into the margins in these circumstances, nor to find marginal notes which might otherwise occur as foot- or even endnotes. Anything which can justify the presence of the anomalous margin is regarded as fair game!

Finally gutters: the internal 'margins' that separate columns from each other in multi-column formats. Generally speaking, a gutter should be no wider than the mean of the left and right margins; if anything, it can be somewhat narrower. Some designers prefer to divide their gutters vertically by a narrow rule; I would tend to avoid this unless rules were used elsewhere in the design. Here, as in many places, the desire for uniformity provides excellent guidance.

2 The elements of a book

Having established guidelines for the overall dimensions of our book, it is now appropriate to consider the various elements which make up that book. At the most superficial level (and ignoring the covers, spine and dustjacket), a book consists of the *front matter* (also referred to as 'prelims'), the *text*, and the *back matter* or *end matter* (the last is clearly ambiguous, as a book has two ends, but traditionally 'end matter' is used in preference to the less ambiguous 'back matter').

The front matter is composed of such elements as the half and full title pages; the copyright and cataloguing-in-publication data page; a table of contents (and sometimes other analogous tables); and perhaps a preface. Also frequently included in the front matter (particularly with the advent of the DTP revolution, since which we have all become far more aware of typefaces and typography in general) is a 'colophon', which strictly speaking should occur as the very last element of the book, but now more usually occupies space on the copyright and cataloguing-in-publication page; the colophon contains details of the typefaces and leading used, and may also give details of designer, printer, etc.

Amongst the end matter are found appendices; one or more indexes; a bibliography (if such is not associated with each chapter, or if an overall bibliography is desired as well as one per chapter); and perhaps a glossary or similar.

Finally, the text is composed of the body of the book; usually divided into chapters, it may also be divided at a higher level into parts.

It is fair to say that the boundaries between these three zones are not entirely rigid: an author may choose to regard a preface as a part of the text, rather than as a part of the front matter, and this will need to be reflected in the page numbering, as we shall see. Similarly some writers may regard their appendices as forming a part of the text; this may affect their page numbering but is less likely so to do. Indeed, an author may choose to write a preface, a prologue, an introduction, a conclusion, an epilogue, and one or more appendices; the designer and author will need to liaise carefully to ensure that each is appropriately classified.

The primary reason for this division concerns page numbering: front matter is traditionally numbered in *roman* style, using lower-case roman numerals (i, v, x, l, c, d, m) which are often set as dropped folios, whilst the text proper is usually numbered using *arabic* numerals (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Appendices and other end matter usually continue in the same sequence and style as the main text, but it is permissible to re-start the numbering for the appendices and prefix the page number with an letter 'A'; if this latter course is taken, the index (assuming that the index forms the very last element of the end matter) will need to have unnumbered pages, as it would clearly be inappropriate to continue using 'A'-style numbering whilst it would be equally inappropriate to resume the main numbering scheme. Fortunately indexes are not required to be self-referential (although I confess to once padding out an index that would otherwise not balance with an entirely spurious reference to 'loop, infinite', whose sole page number was that of the entry for 'loop, infinite' in the index. . .).

There are also conventions as to which elements are required to occur recto, which verso, and which require to be preceded or followed by a blank page. A typical book

might be numbered as followed (remember that odd numbers indicate recto, whilst even numbers indicate verso):

- i. half title;
- ii. blank;
- iii. full title;
- iv. cataloguing-in-publication, copyright, colophon;
- v. preface to the edition;
- vi. general preface;
- vii. ditto, continued;
- viii. blank;
- ix. table of contents;
- x. ditto, continued;
- xi. glossary;
- xii. blank;
1. first chapter.

Of these, the half and full titles are required to occur recto, (whence the blank page between, which also affords a nice contrast to the complexity of the full title page); the copyright and c-i-p page frequently occurs on the reverse of the full title page; the preface is not required to start recto, but it may be the designer's wish that it should so appear; the table of contents is normally recto, as here; the first chapter invariably opens recto, and except in the most casual of styles all subsequent chapters must open recto as well. The page number of the first chapter page could equally well have been '13'; it is a design decision as to whether to continue the numbering sequence from the prelims or whether to start afresh with the main text.

There are fewer conventions concerning the end matter, but it would be normal for the *first* appendix to start recto; subsequent appendices may start recto or verso as necessary; and the index would also normally start recto.

3 Laying out the pages

Although by far the majority of pages in a book are 'normal' pages, it makes a certain amount of sense to start by considering the opening chapter pages, since these contribute a great deal to the book's visual identity and allow a fair degree of artistic licence in their creation. (It is also fair to say that one can waste an enormous amount of time trying to design them!)

When designing one's first book, it is by no means uncommon for people to align the main chapter header (be it 'Chapter 1' or 'Introduction') with the top of a normal page. For some books, particularly those with very short (less than two pages) chapters, this makes enormous sense, for otherwise one can run to far more pages than are strictly necessary (there are also aesthetic reasons why such a design is to be preferred in these circumstances). However, the vast majority of books have chapters whose

page count often runs into double figures, and for such books it is customary (although not essential) to start the opening chapter heading some way down the page. Typically a quarter to a third of the page depth may be reserved for the above-heading space.

There next comes the question of what to put in the heading. If chapters are numbered, one has to decide between 'Chapter 1', 'One', '1' or some similar variant; and if named, whether to also number or just to use the name (and if one uses both names and numbers, then which numbering style to use). It is thought that 'Chapter 1' is a little old-fashioned, but I do not hold to this view. If both numbers and names are used, and if just the arabic number is chosen, then there is also the option of placing the two on the same line, perhaps separated by a colon and the space of the line; if they are put on separate lines, then it is customary for the number line to precede the name line.

Next the question of font: in which font(s) are these headings to appear? In almost all cases, a large bold font will be used, but 'large' is very much in the eye of the beholder; it is probably safe to say that L^AT_EX uses rather larger fonts for this purpose than more conservative designers might choose. The use of a *sans serif* font for such headings is most certainly justifiable, but not essential.

Placement: should the headings be centered or ranged left (or even ranged right)? Generally speaking, centered headings are either slightly old-fashioned or are more suitable for works in the arts; modern scientific publications frequently adopt a ranged-left theme which runs throughout the book, including headings such as these. Ranged-right probably shrieks *avant garde*, but cannot be discounted on that score; if used, there should probably be other elements in the design which echo the ranged-right theme, or there should be a contrasting ranged-left theme to balance. If an epigram is used, it is probably better to have the headings ranged left and the epigram ranged right, as the converse would over-emphasise the epigram to the detriment of the chapter title.

There is another element to placement which also requires discussion: is the white space above the heading to be regarded as belonging to the heading or to the page? By this I mean the following: if the chapter title normally occupies *n* lines (typically one or two), but a pathologically long title for a particular chapter requires one or more additional lines, from where should the space for these lines be taken? Should the title be allowed to extend *up* the page, encroaching on the reserved white space, or *down* the page, displacing the starting point of the main text downwards? Neither is ideal, but if authors insist on writing pathologically long titles, one or other solution must be taken. Although the following is not cast in stone, it is perhaps worthy of consideration: if the opening chapter page starts with a line containing only the *number* of the chap-

ter (or with the word ‘Chapter’ followed by the number), then that should *always* occur in the same vertical position (and thus the main text will get displaced downwards); but if the page starts with the *title* of the chapter, then that title may be allowed to extend upwards, thereby ensuring that the main text always starts at exactly the vertical position on the page.

And rules: should the headings be set off from the text by a horizontal rule? Here we probably need to return to the theme of uniformity: if rules form a recurring theme throughout the book, then a rule between heading and text is probably fine; if not, then it may seem intrusive.

Finally, before leaving the subject of opening chapter pages completely, it might be worth recapitulating on the advice given in the predecessor to this paper concerning running heads and folios: generally speaking, a running head has no place on an opening chapter page; the white space above the title should merge imperceptibly into the top margin. This means that the folio, if normally on the outer edge of the running head, must (on an opening chapter page) either be omitted completely, or must be relegated to the footline. Omitting the folio is highly undesirable, as it renders the table of contents virtually useless (and also reduces the usefulness of the index, if any entry in the index refers to an opening chapter page); the solution is therefore to set the page number as a dropped folio, centered in the footline. Sometimes such folios are given a little additional ornamentation, for example en-dashes on each side set off by a thin space; although this convention is taken directly from typewriter practice it does, in the opinion of the present author, render the folio a little more obvious, and therefore has something to commend it.

Having completed opening chapter pages, the next most significant element in the design of the book is the normal text page; such pages usually make up over 90% of the book, and it is therefore worth expending considerable effort ensuring that they look ‘right’. We have already dealt with margins, gutters, head and footlines, so we may concentrate on the text proper, and in particular on the fonts and leading to be used.

4 Fonts and leading

As suggested above, the text will normally be set in a 10 pt *serif* font, often on a 12 pt leading (here, at least, plain \TeX gives sensible defaults, except in the excessive measure used). There appears to be a widespread belief that Times Roman is the font of choice, yet this font, designed as it was for use in the exceptionally narrow measure of newspaper columns, has little to commend it apart from widespread availability. The font is too narrow for the generous measure of most books, and if it *must* be used can benefit enormously from being anamorphically

scaled by a factor of $24/25$ in the vertical direction. Such scaling, whilst anathema to purists, converts the somewhat narrow letterforms of Times Roman into rounder, softer, shapes, and enables a near optimal combination of font size and leading to be used on measures up to 27 pc and beyond. $11/12.5$ Times Roman, when anamorphically scaled by a factor $24/25$, yields $10.56/12$ which in the opinion of the present author results in a highly readable text.

But far better than anamorphically scaling Times Roman is to select a font which already has the appropriate properties (rounded letterforms, suitability for use with wide measures, etc.); examples are legion, but amongst the most obvious candidates are Baskerville, Bembo, Caslon, Garamond, and Palatino. To be avoided are fonts which are highly idiosyncratic: it is to be remembered that the *sole* purpose of the font is to convey information; if the reader is distracted by the idiosyncratic nature of the font, information transfer will be less than optimal and the book’s value reduced as a result.

It may be worth digressing at this stage to discuss briefly one particular book which I first encountered on being asked to review it, Knuth’s *3:16 Bible Texts Illuminated*. My first reaction on opening this book was to ask myself rhetorically “why on earth did he set it in Computer Modern?”. I was familiar with Computer Modern from the *Computers and Typesetting* quintology, and had, of course, set much of my own material in Computer Modern whilst learning about \TeX ; but I had reached the point where I felt that other fonts had much more to offer, and had not, for some time, typeset anything in Computer Modern at all; it therefore came as a nasty shock to find a book on Bible Study typeset entirely in Computer Modern, particularly by someone whose opinions I value so greatly.

And yet, the strange thing is that having read no more than half a dozen pages of *3:16* I suddenly discovered that I was no longer seeing the font at all; it had, to all intents and purposes, ceased to exist as a typeface, and become purely a medium for the communication of facts. Now Computer Modern, based as it is on Monotype 8a, is not everyone’s ideal font; and particularly when rendered on low resolution devices such as laser printers can be quite unpleasant indeed, with the thin strokes breaking up or disappearing completely and the thick strokes somehow seeming out of proportion. Yet when rendered on a high resolution typesetter, the contrast between thick and thin contributes much to the aesthetics of the font, and the overall effect is to yield an unintrusive design, pleasantly devoid of idiosyncrasies, which suppresses its own personality and allows the information to shine through. Perhaps there is no such thing as a bad font; what we perceive as bad may simply be a good font used inappropriately, or rendered using inappropriate technology.

But to return to the question of design, and in particular to the design of the normal text pages of a book. Having selected our primary font and leading, we will need to select appropriate variants of that font for particular purposes (we may also need to select one or more other fonts for special purposes, but as a general rule the fewer fonts used in a document, the better the document will be). For emphasis, and for foreign words and phrases within the text, it is customary to use an italic variant of the font; the use of bold for emphasis is to be strongly deprecated, with such fonts being reserved for headers and similar. Italics may also be used for book titles, for the names of ships, and for other analogous entities. It goes without saying that underlining, too, has no place in the running text of a book, and very little place anywhere else either; just as the use of bold for emphasis is an artifact of early word-processing systems (which were incapable of italics and therefore had to create an alternative convention for achieving stress), underlining is an artifact of handwritten and typewritten text, and has no place in a typeset document.⁴

If it is necessary to stress a word or phrase within a longer structure that is already being typeset in italics, it is customary to revert to a roman font for the stressed section; but the present author can find no reason why in these circumstances the stressed section should not be set in bold italics, if such a font variant is available (and with the advent of PostScript fonts, such variants are usually to be found); if the bold stressed section is being compared or contrasted with another section of text in the book which is physically nearby, then it may be necessary to set that section too in bold italics, even if it occurs in a context in which italics are *not* being used; in that way, the reader will be given appropriate typographic cues as to which two sections are being compared or contrasted.

Italics (which are a highly stylised variant of a font) should not be confused with *slanted* or *oblique* variants, both of which involve no original design but result from a simple geometric transformation of the roman form of the font. Whereas italics and oblique forms both have an honourable ancestry (oblique normally being reserved for *sans serif* fonts whilst italics are normally a variant of a *serif* form), slanted fonts appear to be another artifact of the DTP revolution. In the opinion of the present author they have little to offer in the way of aesthetics, and even though they are sometimes used where it is deemed desirable to differentiate typographically between two entities which would otherwise both have to be rendered in italics, as a general rule I would caution against their use. Designers have managed for centuries to convey considerable amounts of information without having recourse to slanted fonts; it is to be hoped that future generations of designers will conclude that they represent no more than what Fowler might have termed ‘elegant vari-

ation’, and are therefore a luxury without which we can all happily do.

It is sometimes necessary, particularly in books on linguistics or other subjects in which language is both used and discussed, to differentiate typographically between the two uses. Sometimes simple quotations marks will suffice; sometimes italics; but there are also times when both of those forms are already reserved for other typographic differentiation, and some third form is needed to clarify which text is being discussed and which text is performing the discussion. In these circumstances (and in very few others), it is justifiable to introduce a new font which may be used as a part of the running text. If the main text is set in a *serif* font (as it almost invariably will be), then a second *serif* font would *not* be suitable; even though two *serif* faces may be as different as chalk and cheese, the risk of confusion is still too great (and the aesthetic clash too severe) to permit two distinct *serif* faces to appear in juxtaposition. The second font must therefore be a *sans serif* face, chosen to blend in with, whilst being clearly differentiable from, the main text face. The second font will need to be matched for weight (visual density), ex-height and caps-height; and because of the variation in the semantics of *design size*, will probably need to be loaded at a fractional size.

5 Headings

The motto for the predecessor to this paper was “There can never be too little space below headings, only too much!”, and in those few words can be summarised the bulk of the received wisdom concerning headings. As previously pointed out, a heading *must* be tied to the text with which it is associated, and that text is invariably the text which immediately follows. Headings are frequently hierarchical in nature, and lower-level headings are more closely bound to the following material than higher-level; thus the white space which separates low-level headings from the text is usually less (and never more) than the white space which separates higher-level headings and text. In the limiting case, the heading is *run in*, that is to say literally forms a part of the text and does not occupy a line in its own right. For run-in headings, it is essential that the author be consistent in usage, since such headings can either participate in the grammar of the text or remain a distinct

4. Of course, like almost every rule, these rules too admit of exceptions, and it would be a brave author indeed who wrote that *every* instance of underlining, or of the use of bold within running text for emphasis, was categorically wrong; the most that can be said is that generally speaking such (ab)uses are regarded as infelicitous or inappropriate, and that should the designer none the less decide to adopt such a convention, he or she should be aware of the ‘rules’ that are being flouted, and take a conscious decision to flout them rather than simply being unaware of their existence.

grammatical entity; in the former case it is customary to indicate the extent of the heading by a change of font (italics, or bold, or even caps and small caps), but by no extra horizontal white space or punctuation; for headings which are grammatically distinct from the text which they introduce, a change of font is also indicated, but punctuation (e.g. a colon) or additional white space (e.g. one quad) is also frequently used. Such a heading might be set off by as little as 1 ex additional white space from the preceding text, and certainly by not more than one blank line.

At the next level in the hierarchy, the heading usually occurs on the immediately preceding line, and occupies a line in its own right. It is not set off by any additional vertical white space, but simply separated from the text by the normal leading for the paragraph. Again a change of font is indicated, and the font options applicable to run-in heads are equally applicable here, although the use of caps and small caps would be unusual. The extra vertical white space above the heading is of the same order of magnitude as for run-in headings.

A level higher and perhaps a larger font is indicated. Assuming a base setting of 10/12, a 12 pt font might be suitable for such a heading. If a bold font has been used for any lower level, then this font too must be bold, otherwise ambiguity will result (the same is true at *all* levels in the hierarchy: once a bold font has been used at a lower level, bold fonts must be used at all higher levels. In the same way, no font used in a higher level heading may be smaller than a font used in a lower level heading; it may be the same size, but only if it is bold and the lower level is not, or if there is other clear typographic indication of the hierarchy). Above such a heading a little extra white space might be allowed, perhaps between one and one-and-a-half blank lines.

Beyond this point, simple extrapolation is sufficient: as we move up the hierarchy, headings get bigger, bolder, more distinctive. The white space below them may increase, but only very slightly; the white space above increases, but not to ridiculous limits. Anything in excess of three blank lines is almost certainly excessive, and two blank lines are normally more than sufficient.

At this point it is appropriate to consider the implications of the above set of rules on T_EX implementations. In order to allow successful page makeup in T_EX, it is customary to allow the vertical white space associated with headings to be flexible (i.e., ‘rubber lengths’, in L^AT_EX’s quaint terminology); but T_EX has two quite distinct rules when dealing with flexibility: if a dimension is given a negative flexibility (i.e., is allowed to shrink), then T_EX will take advantage of the stated shrinkability if necessary to achieve optimal page makeup, but will never attempt to shrink it by more than the permitted amount; however, if a dimension is given *positive* flexibility (i.e., is allowed to stretch), then T_EX will first of all take advantage of that flexibility

to achieve optimal page makeup, and if that flexibility is insufficient, *will continue to stretch it until optimal page makeup has been achieved*, even if this involves stretching it by many times its stated stretchability. Of course in these circumstances T_EX issues a warning, but by then it is too late: the evil deed has been done.

The implications of this behaviour for successful implementations of design are quite severe: T_EX must *never* be given positive stretchability to use if it is required to exercise any automatic control over the upper bound by which white space will be stretched; shrinkability can be used, but T_EX is noticeably asymmetric in this respect, and whereas `\vfil` and its friends can be used to pad out underfull pages whilst preventing embedded ... plus \$n\$ pt constructs from contributing white space, there is no equivalent which can be used to negatively pad pages whilst preventing ... minus \$n\$ pt constructs from shrinking (the reason is that T_EX will not allow what it terms ‘infinite glue shrinkage’ to occur in unrestricted horizontal or vertical modes). Thus there are severe problems in inhibiting T_EX from taking excessive advantage of permitted flexibility, and in the end only careful observation of the log file, and manual intervention where T_EX has exceeded its brief, will be sufficient to keep matters under control.

But recalling for a moment the discussion on grid-based layouts which took place in the predecessor to this paper, it will be appreciated that simply preceding and following header lines by `\vskip` commands will not necessarily have the desired effect. A far more satisfactory method of placing headers, whilst ensuring that they occupy an integral number of blank lines (i.e. an integral multiple of `\baselineskip`) relies on a technique which I refer to as a ‘pseudobox’: this is a T_EX construct which is in reality a box whilst behaving like glue; the following code fragment illustrates the technique in use.

```
\newbox \headerbox
\newdimen \headerheight
\newdimen \headerdepth
\def \header #{\afterassignment \afterheader
\setbox \headerbox = \vtop}
\def \afterheader {\noindent
\aftergroup \reallyafterheader}
\def \reallyafterheader
{\headerheight = \ht \headerbox
\headerdepth = \dp \headerbox
\advance \headerheight by \headerdepth
\headerdepth = \headerheight
\ht \headerbox = 0 pt
\dp \headerbox = 0 pt
\advance \headerheight by 0.5\baselineskip
\divide \headerheight by \baselineskip
\multiply \headerheight by \baselineskip
```

```

\ifdim \headerheight < \headerdepth
  \advance \headerheight by \baselineskip
\fi
\vskip 0 pt
\box \headerbox
\vskip \headerheight
\noindent
\ignorespaces
}

```

If this code is used to typeset a large bold header within the text of this paragraph, as in `\header {\Huge Header}`, the effect *should* be to leave the remainder of the paragraph set on its natural grid;

Header

whether or not it has achieved this effect is left to the reader to see! Perhaps a brief explanation of the code is in order, as so far as the author is aware the technique has not previously been published. The `\header` macro takes no parameter, but the terminal hash of its parameter list causes it to require an open brace to immediately follow its use; on the assumption that the open brace is the open brace of a brace-delimited parameter (which it should be, if the macro has been properly used), the macro sets `\headerbox` to a `\vtop` containing the parameter. However, an additional token is introduced into the `\vtop` just prior to the parameter by means of the `\afterassignment`, that token being `\afterheader`. This token itself expands into three further tokens, `\noindent` (to prevent the parameter from being indented within the box), `\aftergroup` (to allow the following token to be expanded not within the box but outside it, once it has been set), and `\reallyafterheader`, which is the macro that does all the real work. Thus the combined effect of the `\afterassignment` and the `\aftergroup` is to inhibit any indentation of the parameter, and to cause `\reallyafterheader` to be expanded once the box has been set. `\reallyafterheader` commences its work by saving the height and depth of the box in which the header has been set, and then computes their sum; the height and depth are set to 0 pt. Using Knuth's algorithm from A15.8, the combined height + depth is rounded to the nearest integral multiple of `\baselineskip`, and if the result of this rounding is less than the original sum, a further increment of `\baselineskip` is added. The result of this computation is the smallest integral multiple of `\baselineskip` within which the entire contents of the box can be set. A vertical skip of 0 pt is carried out (to force `TEX` into vertical mode), and then the box is typeset (remembering that it has zero apparent height and depth, and therefore occupies no space), after which a further `\vskip` of the calculated integral multiple of

`\baselineskip` is carried out to leave room for the contents of the box whilst not disturbing the regularity of the baseline grid. Finally `\noindent` and `\ignorespaces` ensure that the first paragraph following the header is typeset correctly.

A real-life instance of this code would require parameterisation to indicate the level of header, from which it could ascertain (by means of a look-up table) how to distribute any required additional space around the header; in addition, it would enable ragged-right setting within the header box, and would need to deal correctly with a header immediately followed by another header (the spacing should not be additive). Many other refinements are possible.

6 Paragraphs

In trying to make practical recommendations for real-life book design, it is necessary to alternate between those entities which occur fairly rarely (opening chapter pages, headers, etc.) and those which form the bulk of the book (regular pages, paragraphs, etc). Here we consider material which makes up the vast bulk of the book, to wit the paragraph.

Fortunately the 'rules' for paragraphs are fairly straightforward, but as so many examples may be seen which either blatantly ignore the rules or are simply unaware of them, some discussion is none the less necessary. It should be noted, however, that these rules are inherently culturally based, and I am advised by one eminent French authority⁵ that the rule stated below concerning the first paragraph of any new section would be incorrect were it to be applied to material published in French.

- The first paragraph of a new section is not indented. This rule is so often more honoured in the breach than in the observance that I sometimes wonder whether its existence is widely known at all. For reasons entirely unclear to me, `LATEX` whilst doing its best to honour this rule indents abstracts, which seems to me at best inconsistent and at worst inexcusable. I am very pleased to see that these proceedings avoid that error.
- A paragraph is either indented, or is set off by vertical white space from preceding material. It is normally considered infelicitous to do both; it is a gross error to do neither. The reason why the latter is so severe a crime is that if paragraphs are neither indented nor set off by vertical white space, then any text in which a paragraph just happens to end flush with the right margin will be followed by a paragraph whose existence can barely

5. Bernard Gaulle, past and future President of GUTenberg, the French-speaking `TEX` Users' Group.

be guessed at. There will be *no* typographic clues to indicate that a new paragraph has started.

- The leading and font within a paragraph are uniform. This may seem to go without saying, but if a document is set with the minimum leading necessary for unadorned text, then an accented capital letter may well be enough to force down the entire line on which it occurs. In such circumstances either the leading must be increased for the entire document, or special steps taken to conceal the height of the accented letter (whilst ensuring that it does not unfortunately co-incide with a descender from the line above). By ‘uniform’, when applied to the font, I do not suggest that every glyph in the paragraph must be set in the same font; clearly there may be a need for italics, or even for a *sans serif* font at points, as indicated above. But all the glyphs within the paragraph should *appear* uniform, and must therefore come from closely related or well chosen fonts. For example, the first phrase of each paragraph in a book may be set with an initial full cap and then small caps; provided that these blend in with the main text font, there can be no objection to this. Similarly the first letter of the paragraph may be a dropped cap; provided that it too blends in with the main text font, that is a perfectly valid design decision (and sometimes very stylish, if I may express a personal opinion).
- A paragraph should not end with only a part-word on the last line. Assuming that hyphenation is permitted at all (which it will need to be if fully justified text is specified), then the last line of a paragraph should end with at least one full word and preferably more. Plain T_EX’s (and L^AT_EX’s) setting for `\parfillskip` do not encourage this; a more felicitous setting might be `\parfillskip = 0 pt plus 0.7\hsize`, which encourages longer last lines at the expense of setting some such lines slightly loose.

7 Graphics, figures, and other ‘floating’ entities

Although there is much more that can (and should) be said about book design in general, I feel that there is one area which must be treated before I close, and that is the whole area of insertions, or as L^AT_EX terms them, ‘floats’. These are, in some general sense, graphic entities, although they may turn out to be purely textual in content. What really typifies them, however, is that they are invariably indirectly referenced; that is, they are referenced by the author in terms of *see Fig. 1* or *See Table 2.4*, rather than being implicitly referenced by position in the text as in, for example, *as shewn below*. By virtue of the indirect nature of their reference, they can be physically remote from the point of reference, but one of the major skills of page

makeup is the careful placement of such entities. The cardinal rule for these insertions is that *they must be capable of being seen from the point of reference*. One of the little appreciated strengths of T_EX is how well it carries out this task for footnotes, which are a very simple instance of insertions; if you look carefully at T_EX-set material which has many footnotes, you will probably be surprised at the number of times that a footnote reference occurs on the very last line of the page (before the footnotes themselves appear, that is). If you have not thought about this problem before, you may casually remark to yourself “that’s lucky; another line and the footnote marker and its text would have appeared on different pages”. But now try to find an instance where that has happened; try as you might, I suggest that you won’t. And that surely suggests that it is more than luck that causes that particular juxtaposition of footnote marker and start of footnotes to occur so regularly, so reliably, and so consistently. And of course it *is* more than luck; all the while that T_EX is accumulating material in galleys, it is carefully tracking how much space is occupied by footnotes and how much by the main text; and as soon as the combination of the two exceeds the available space on the page, T_EX knows that it must cut the galley at or near that point and start a new page.

Now footnotes are, as I said, a particularly simple instance of such insertions; no-one minds if the text of a footnote is started on its page of reference but continued on the next (no-one but a pedant, that is). But figures, tables, graphics, etc., are a very different kettle of fish; they are essentially indivisible entities, and can therefore either appear on a given page or not appear on that page; there are no half measures which would allow a part of the figure/table/graphic to appear, and the remainder to appear on the next page.

So now put yourself in the position of T_EX, this time not accumulating text and footnotes, but accumulating text and (say) figures. T_EX continues to accrete material in its galley as before, and encounters a reference to a figure; say that the page is only a third full. If the figure is less than two-thirds the depth of the page, there is no problem: T_EX simply adds the figure to the list of things that appear on that page and carries on. But now let there be a second figure reference, maybe two-thirds down the page: T_EX looks to see how big the figure is, and discovers it needs a half a page to itself. What does T_EX do? The first choice is trivially ruled out; you can’t have the reference to the figure followed by the figure, because (a part of) the figure would fall off the bottom of the page. OK, what’s the next choice? Remember that the figure can *float*. So, let’s try floating the figure to the top of the page on which it was referenced: no problem there, the figure appears at the top of the page, pushing the textual material material down. Some

of the textual material will fall off the bottom of the page, of course, because we already know that we have $2/3$ of a page of text, and $1/2$ of a page of graphics, so $1/6$ of a page of text falls off the bottom. But that's no problem, because textual material can normally be split at almost any point: so $\text{T}_{\text{E}}\text{X}$ chooses the nearest valid breakpoint and carries the remaining material over to the next page.

Then what happens? Well, think about what is *on* the material that has been carried over: the reference to the figure that caused the trouble in the first place! So now we have the figure on page n , and the reference to the figure on page $n + 1$. If $n + 1 \equiv 1 \pmod{2}$ (sorry, if $n + 1$ is odd!), then there is no real problem, for the reference to the figure occurs on the recto half of the spread, and the figure itself occurs on the verso half of the spread, so all is well. But if $n + 1$ is *even*, then all h@ll breaks loose: because the figure is on the recto half of a spread, and the reference to the figure is on the verso half of the next spread; and when the reader finally encounters the reference to the figure, the

figure itself can no longer be seen. And no matter what $\text{T}_{\text{E}}\text{X}$ were to do in those circumstances, it would not be able to solve the problem without assistance.

So there are some problems in page makeup that simply *cannot* be solved by naively applying rules; rules are all very well, but eventually the time will come when the author's text and the rules of design are simply incompatible, and in those circumstances you will have little option but to liaise with the author and attempt to persuade him or her to re-write the offending portion of the text. If the author is dead, and the text is cast in tablets of stone, then you will have to do a lot of work by hand, maybe setting a whole series of paragraphs one line looser than ideal, just to force a reference onto a more appropriate page. But when you've done it, and the finished book is printed, and you look at it and *know* that there are no further improvements that you could have made, then a great warm glow will fill your body and you'll know that it's all been worthwhile. Good luck!

Bijlage 5

Tiling in PostScript and METAFONT – Escher's wink

Kees van der Laan

abstract

Drawing tilings by computer is discussed. Examples are borrowed from literature. New are their included METAFONT and PostScript programs, with sometimes a new variation of a picture.

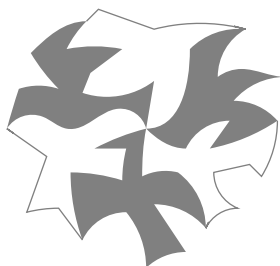
keywords

Apollonius, Computer art, Douat, Dürer, education, EPS, Escher, graphics, METAFONT/Post, outline, Kepler, Mondrian, parquet, Penrose, PostScript, puzzles, Soddy, tiling, Truchet, turtle graphics, zero finding.

1 Introduction

Children like to play, like jigsaw puzzles. Tiling avant la lettre, where the form and color of the pieces matter.

Example (*Part of Escher's Sun and Moon*)



Dark birds in bright daylight or white birds in the dark?

What has this got to do with tiling?

Well ... it is a tiling of free forms. When considered as a non-tight tiling of either dark or white birds the others are spurious. Intriguing. Underneath is a grid of triangles, reshaped into birds.

Escher anthropomorphised tiling by introducing creatures as decoration of the tiles as opposed to traditional tiling. Tiles used to take strict geometrical patterns especially the Islamic ones because of religion.

Escher considered lines as to be 2-sided, and exploited this in depth. Seen from one side the line is the boundary of a figure and looked upon from the other side gives meaning

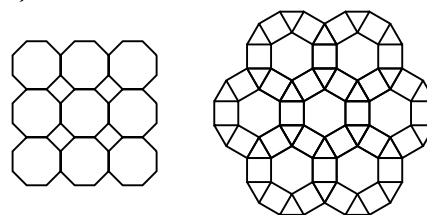
to another figure.

Tiling has all to do with filling up the plane. It is an old human activity of which beautiful results have been preserved, for example in the Alhambra.¹ In this note a few examples of tilings are shown next to their codes – METAFONT, or POSTSCRIPT – for drawing them by computer.

Intriguing and captivating are Escher's contributions. In some patterns he not only translates but also rotates and reflects. In others, the so-called limit cycles, he shrinks the size of the basic element gradually, suggestively infinitely. In his metamorphoses tiles change their form in crescendo.

Example (*Classification of homogeneous tiles*)

The following homogeneous, Archimedean tiles are classified in mathematical tiling theory by the polygons at each vertex. The left is denoted by $\{4, 8, 8\}$ – a square and 2 octagons join at each corner – and the right one by $\{3, 4, 6, 4\}$.²



However, no attempt will be made to treat mathematical tiling theory. It seems that a constructive computer-oriented approach and programming terminology are needed. I found it advantageous to concentrate on non-tight tilings with spurious elements. For example the right figure – extended infinitely – can be drawn by just squares appropriately positioned. The result will contain the spuri-

1. A thorough reference about tiling is: Tilings and patterns by Branko Grünbaum & G C Shephard. Freeman, New York. ISBN 0-7167-1193-1. Martin Gardner in his Scientific American notes has popularized various (mathematical) puzzles. Bouwkamp, a Dutch physicist, has worked on 3D tilings. MacGillavry, a Dutch crystallographer, has analyzed Escher's use of symmetries and correlated this to crystal symmetries. Coxeter, a Canadian mathematician, has worked with Escher and contributed to mathematical tiling theory, if not for the limit cycles.

2. Do you recognize a variant of the Pythagorean tree in the right one? Isn't it amazing? Pythagorean trees will be treated in the note on fractals.

ous triangles and hexagons.

To draw Pythagorean trees needed a lifetime in the 40-ies. I don't know how much time it took to draw tilings, but with the computer it is just a matter of programming, which is substantial less especially when we can build from templates.

1.1 Coding

The coding of tilings comes down to finding a basic element—a tile—and to make compositions of rotated and/or translated copies. Because these symmetry operations can be easily programmed for a computer, drawing and designing tilings can be seen as computer art nowadays.

In contrast with general POSTSCRIPT codes which are usually generated by programs just for use, my codes are concise, consistent,³ educational, procedural, and next best to literate.

The POSTSCRIPT codes are ready for use. The METAFONT codes don't build a character, just the inherent aspects are shown, to illustrate differences with coding in POSTSCRIPT.⁴ For use adaptations are needed reflecting your computing environment, be it to build a character from the code and export this etc., or to make a MetaPost code from it, and so on.

Not all codes have been included, especially the more elaborate ones, mostly of composite tilings, have been omitted. Nor are there METAFONT codes for all pictures. The reason is that I started with METAFONT but experienced later on POSTSCRIPT, and when the note grew only POSTSCRIPT codes were developed, which for these problems is sufficient and practical enough, given my situation. It is hoped that the included METAFONT codes serve their purpose.

Fractal tilings and space filling curves—well ... mathematical curves—will be treated in separate notes.

All the enclosed pictures have POSTSCRIPT codes; none has been scanned.

Conventions I've adopted the following conventions in coding POSTSCRIPT and hope by stating them it will ease the reading of the codes.

- Constants are at the beginning and have short names. The prefix *m* (minus) denotes the negative value, *h* denotes half the value. The postfix *loc* denotes a quantity to be used locally, especially in recursion. So *mhsloc* denotes $-.5s$, to be used locally.⁵
- */name* *exch* *def*, lays hands on the value on the top of the stack. This is used to reuse an argument (by name), supplied on the stack (before the invoke of an operator).

Generally, intelligibility is served by appropriate spacing, and when spacing is absent either it is trivial or spacing is

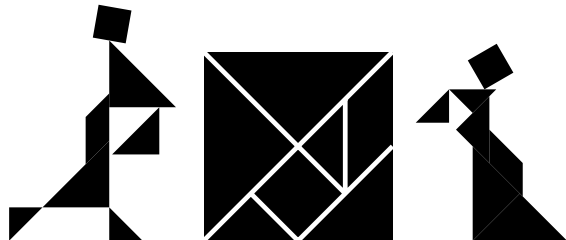
not relevant for understanding.

The following selection of structuring elements are reminders for those not familiar with POSTSCRIPT.

- definitions⁶
 - /name... def*
 - /name{...} def*
- structures
 - gsave... grestore*, the graphics state delimiters⁷
 - boolean {truepart} if*
 - boolean {truepart} {falsepart} ifelse*
 - value {loopbody} repeat*
 - begin step end {loopbody} for*
 - array {loopbody} forall*⁸
 - {loopfirstpart boolean {exit} if loopsecondpart } loop*

Example (Tangram)

Another way of looking might emerge from playing with Tangram-like puzzles, to combine the (holy) 7 pieces into free forms, stimulating phantasy.



In POSTSCRIPT the Tangram tablet can be drawn as follows.

```
%!PS-Adobe- Tangram, cgl Jan 97
%%BoundingBox: -50 -50 50 50
/s 50 def      /ms  s neg def
/hs s 2 div def /mhs hs neg def
ms ms moveto ms  s lineto
 s s lineto s ms lineto
closepath fill %square
s s moveto ms ms lineto
```

3. I did not pluck from the net codes programmed by various people, with as a consequence that the programs enjoy a common approach and programming style facilitating intelligibility.

4. My METAFONT codes are essentially compatible with MetaPost codes, because I pay attention to use only the common features. For example, I don't rely on the fact that in METAFONT pixels can have more than two values.

5. METAFONT allows $-.5s$ as such, nice syntactic sugar.

6. The difference is that in the first only the result is associated and stored under the name, while the latter stores and associates all that is between the curly braces with the name.

7. There are no scope braces.

8. For all the values supplied in the array the loop is executed.

```
mhs mhs moveto 0 ms lineto s 0 lineto
1 setgray 3 setlinewidth stroke
ms s moveto hs mhs lineto hs hs lineto
2 setlinejoin stroke %showpage
```

The `2 setlinejoin` prevents unwanted sharp corners. If you understand this code this note should be easy reading for you, and do contribute gems of your own.⁹

1.2 Audience

The aimed at audience consists of users of (La)TeX, METAFONT/Post, POSTSCRIPT, ... who are familiar with programming and not afraid of coding in terms of graphical primitives. The benefit of straight POSTSCRIPT (hand)coding is conciseness, efficiency, portability and universality. The drawback is little assistance, scarce diagnostic reports, tedious proofing, and so on, while developing the codes. Maybe the included codes can function as templates.

Reading (and understanding) all is too much hoped for. Nowadays we are flooded by information, and we all seem to suffer from the disease of our times: lack of time. If only *one* example will appeal to you or spark your imagination, I'll be happy.

1.3 Why?

What has POSTSCRIPT got to do with TeX?

EP history has it that TeX and POSTSCRIPT form a real good team. POSTSCRIPT provides the graphical primitives which TeX is lacking.

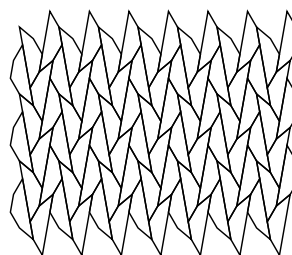
Some authors provide codes in BASIC, such as Lauwerier and Peitgen c.s. I completely agree that for those cases where images can be specified in terms of just a few graphical primitives it is wise to use omnipresent, cheap and trustworthy tools like BASIC (and Ghost/POSTSCRIPT), especially when time-invariance of the algorithms is at stake. However, with the ubiquitous TeX together with multiplatform drivers which so easily and seamlessly merge the POSTSCRIPT pictures into the dvi script, one can go a step further and use POSTSCRIPT instead of BASIC, the more so because I expect POSTSCRIPT to enjoy a significant lifetime.

The above does not hold for other proposed teams like TeX and SGML or PDF (yet) IMHO, with all respect.

1.4 Division of the plane

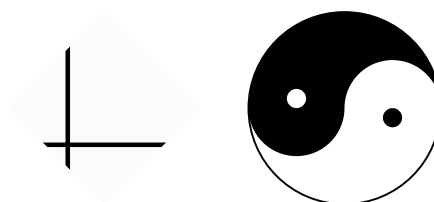
Related to tiling is the art of cutting a figure into pieces. And if we consider colors the stained-glass windows come to mind as beautiful examples.

Example (Stained-glass (monohedral) pattern)



In black-and-white the following are simple but famous examples.¹⁰

Example (Lines and arcs)



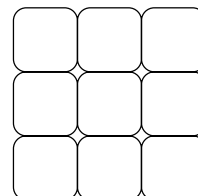
Puzzles bridge the gap between cutting a picture into pieces – the design – and tiling – the remake.

2 Squares

A square is a common tile. An ordinary pavement is built from squares. Bathroom walls are tiled with them. And so on. Empty squares with deformed sides are abundant.

It's amazing what can be achieved with just squares. Cut out the next one and tile, or better still switch on your computer and try some of your own.

Example (Squares with rounded corners)



The coding below illustrates how to tile in POSTSCRIPT: create a (path for a) tile, symmetrically around the origin,

9. In order to draw (free) forms it is better to name the pieces – triangle, square and diamond – and manipulate these. However, it is rather cumbersome to position them into patterns. For puzzling purposes it would be better if the X-works software provide the tangram pieces as part of their icons or allow a user to add icons. Considering the total process especially the reuse – in papers, as logos and so on – I'm happy with the investment of having coded the pieces in raw POSTSCRIPT.

10. For the codes see the Appendix.

and draw translated copies.

```

%!PS-Adobe- Tiling of rounded squares
%%BoundingBox: -75 -75 75 75
/r 10 def /a 50 def /ma a neg def
/ha a 2 div def /mha ha neg def
/tile{4{ha r sub mha moveto
      mha mha mha mha r add r arcto
      90 rotate
    }repeat
}def
ma a a{/i exch def
ma a a{/j exch def
  gsave i j translate tile stroke grestore
}for}for %showpage

```

Remarks. Programming the rounded corners via an appropriate use of `arcto` is borrowed from Adobe's red book, the POSTSCRIPT reference manual. There are two parameters: the side of the square and the radius of the rounded corners. Note how the values of the hidden loop variable can be used.¹¹

In METAFONT the coding of the tiling template can read as follows.

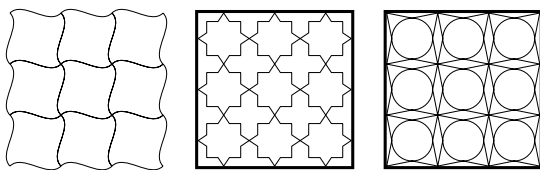
```

a=50;r=5;
path p,tile; pickup pencircle scaled 1;
p:=quartercircle scaled 2r
  shifted (.5a-r,.5a-r);
p:=p--(p rotated 90);
tile:=p--(p rotated 180)--cycle;
for i=-a step a until a:
for j=-a step a until a:
  draw tile shifted(i,j);
endfor; endfor; showit; end

```

Remarks. Another possibility is to draw the tile and copy pictures. Note that only the corners are specified; the connecting straight lines are implicit.

Example (*Variant tiles*)

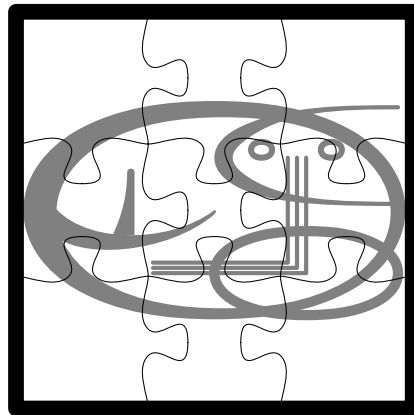


In art we have the tilings by members from 'De Style' like Mondrian and van Doesburg. But before we go over to Mondrian we will play a little longer with nearly empty tiles.

Example (*Puzzle*)

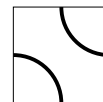
I leave it to your imagination how to draw the enclosed

puzzle. It is fun to code the puzzle pieces. I did it with only two pieces, which of course are each others inverses in the sense that the sides fit. Nice exercise in using POSTSCRIPT's symmetry operators. For the background any picture would do of course.

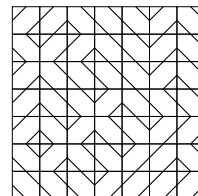


Example (*Truchet tilings*)

Instead of deterministic tiling we can select tiles randomly from a set and tile with these. Truchet¹² considered the tile, well...together with its rotated variant, to yield a set of two.



Patterns composed of these resemble the dragon figures as mentioned in The T_EXbook. In the following the tile is even further simplified by replacing the quarter circle by a straight line.



The above is obtained by the following code.

```

%!PS-Adobe- Truchet's tiling, cgl Mrt 96
%%BoundingBox: -87.5 -87.5 87.5 87.5
/a 25 def /ma a neg def
/ha a 2 div def /mha ha neg def
/tile{rand dup 2 idiv 2 mul eq {90 rotate}if

```

11. PostScript's 'for' pops the value of the loop control variable on the stack among other things, such as executing the loop body. 'i exch def' uses the value on the top of the stack and stores the value-name pair in the current dictionary, ready for later use of the loop control variable value by the name i.

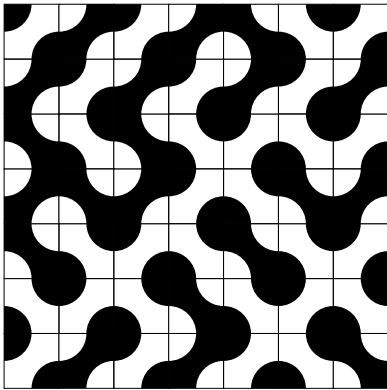
12. A French dominican priest of the early 18th century.

```

mha mha moveto ha mha lineto
ha ha lineto mha ha lineto
closepath .1 setlinewidth stroke
contents
}def
/contents{0 ha moveto ha 0 lineto
          0 mha moveto mha 0 lineto
          1 setlinewidth stroke
}def
/dotiling{f ma mul a f a mul{/i exch def
          f ma mul a f a mul{/j exch def
          gsave i j translate
          tile stroke grestore
          }for}for
}def
%
/f 3 def 5 srand dotiling showpage

```

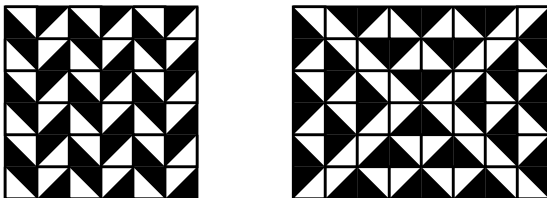
The GUTenberg (French) TUG had in 1995 a contest along with their PSTricks/POSTSCRIPT tutorial about tiling à la Truchet.¹³ It turns out that for coloring 2 colors are sufficient. Have a try.¹⁴



A nice game. Of course the curves can be varied further and nice results will come your way.

Example (Douat's parquets)

A variant for parquets where a square is divided into 2 along the diagonal. By properly arranging the elements parquets are obtained. Douat¹⁵ classified varies parquets.



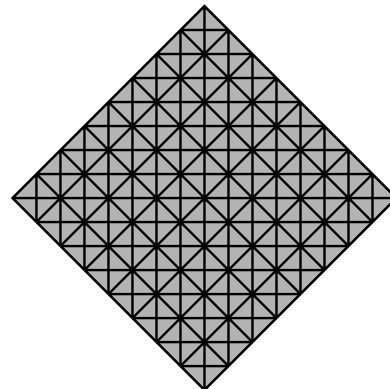
The left figure is obtained as follows.

```

%!PS-Adobe- Douat's parquets, cgl April 97
%%BoundingBox: -37.5 -37.5 37.5 37.5
/a 25 def /ma a neg def
/ha a 2 div def /mha ha neg def
/qa ha 2 div def /mqa qa neg def
/ll{gsave qa mqa moveto
    mqa mqa lineto
    mqa qa lineto closepath fill
    qa mqa moveto
    qa qa lineto
    mqa qa lineto stroke
    grestore
}def
/lr{gsave 90 rotate ll grestore}def
/ur{gsave 180 rotate ll grestore}def
/ul{gsave -90 rotate ll grestore}def
/tile{gsave mqa qa translate ll grestore
      gsave mqa mqa translate ur grestore
      gsave qa mqa translate lr grestore
      gsave qa qa translate ul grestore
}def
ma a a{/i exch def
ma a a{/j exch def
      gsave i j translate tile grestore
}for}for %showpage

```

Example (Lozenge by Mondrian)



13. As communicated by Denis Roegel, see Cahiers GUTenberg 1995 for details. For solutions and the winner see the spring 1997 cahiers.

14. I solved the problem in POSTSCRIPT by looking at the colored pieces as pieces of a puzzle to be put together under the restriction of continuity of the colors along the sides. I grouped the 4 tiles into 2 sets. The tiling is obtained by taking a tile from each group on turns. Which tile to take from a group is free and therefore can be subject to throwing dice.

15. A French priest of the early 18th century. See H.A. Lauwerier(1988): Symmetrie-Regelmatische structuren in de kunst. Aramith. ISBN 90 683 4032 8, NUGI 819. As usual it contains also programs in BASIC

Remark. The real thing is not so perfect, it has lines of varying width making it more interesting. For computer art it might be a challenge to imitate this. But maybe we should leave the imperfections to humans and concentrate on what the computer is good at.

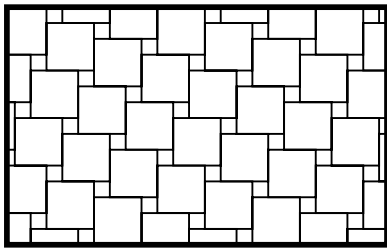
The POSTSCRIPT program below illustrates another approach to coding when the pattern is so straight: (redundant and clipped) lines, no tiles. Note the use of the symmetry by rotating over 90 degrees.

```

%!PS-Adobe- Mondrian's Lozenge, cgl Jan 97
%%BoundingBox: -80 -80 80 80
/e 10 def /r e 8 mul def /mr r neg def
/frame{r 0 moveto 0 r lineto mr 0 lineto
  0 mr lineto closepath
}def
/lines{-7 1 7{e mul dup mr moveto
  r lineto
}for stroke
}def
/diagonals{2 2 15{e mul dup r moveto
  neg mr exch lineto
}for stroke
}def
gsave
frame gsave .95 setgray fill grestore clip
lines diagonals 90 rotate lines diagonals
grestore frame stroke %showpage

```

Example (Van Doesburg inspiration)



```

%!PS-Adobe- Van Doesburg squares, cgl Feb 97
%%BoundingBox: 25 -6.25 250 115
/s 25 def /ms s neg def
/alfa s 3 div def /malfa alfa neg def
/hs .5 s mul def /mhs hs neg def
/square{hs mhs moveto hs hs lineto
  mhs hs lineto mhs mhs lineto
  closepath stroke
}def
/frame{s mhs moveto 9 s mul mhs lineto
  9 s mul 4.5 s mul lineto
  s 4.5 s mul lineto closepath
}def
gsave frame clip

```

```

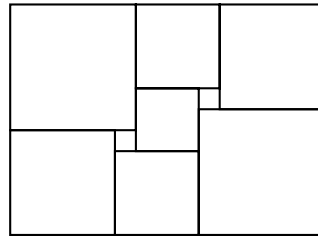
11{gsave 11{square s malfa translate}repeat
  grestore alfa s translate
}repeat grestore
frame 3 setlinewidth stroke %showpage

```

Explanation. It is all about one square with appropriately shifted copies, with the shift parameterized by α . Variant patterns can be easily obtained by changing the value of α . The small squares are spurious. In reality van Doesburg enriched patterns like these by coloring the squares.

Example (Squaring the square)

Is it possible to partition a square into unique squares? A simpler problem is to divide a rectangle into squares, and allow double occurrences.¹⁶



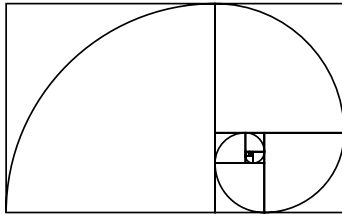
The following is a straightforward POSTSCRIPT code, with the use of (point) inversion. Interesting, and not trivial IMHO, is a to write a POSTSCRIPT operator to draw the picture automatically starting with Bouwkamp's notational array as argument. A nice application of nested forall-s. For the case at hand the notational array reads [[6 4 5] [3 1] [6] [5 1] [4]].

```

%!PS-Adobe- Squaring the square, cgl Mrt 97
%%BoundingBox: -75 -55 75 55
/square{%s(ide) on stack
  /s exch def
  /hs .5 s mul def /mhs hs neg def
  mhs mhs moveto hs mhs lineto
  hs hs lineto mhs hs lineto
  closepath stroke
}def
30 square
2{gsave -45 25 translate 60 square grestore
  gsave 5 35 translate 40 square grestore
  gsave 50 30 translate 50 square grestore
  -1 -1 scale
}repeat %showpage

```

¹⁶ For an account of the squaring the square problem see for example Gardner's More Mathematical puzzles booklet.

Example (*Nested rectangles and spiral of life*)

The nested rectangles have as ratio between width and height of the sides .618, that is an approximation of the golden ratio constant, commonly denoted by ϕ .¹⁷ Nice is the (approximate) life spiral curve through the corners of the squares.

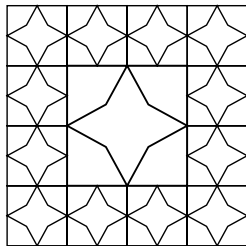
```

%!PS-Adobe- Shrinking squares, cgl Feb 97
%%BoundingBox: 0 0 200 125
/x 200 def /y .618 x mul def
/square{0 y lineto y y lineto y 0 lineto
  y 0 y 180 90 arcn y y translate
}def
12{0 0 moveto square -90 rotate
  /aux x def /x y def /y aux y sub def
  }repeat stroke %showpage

```

Example (*Icons*)

A central painting surrounded by smaller ones – in Russian: klema – illustrating episodes related to the central image.



In the code below the peripheral squares are the same, but can be varied of course.

```

%!PS-Adobe Icon, cgl Feb 97
%%BoundingBox: -64 -64 64 64
/s 64 def /ms s neg def
/hs .5 s mul def /mhs hs neg def
/element{%hs on stack
  /hsloc exch def /mhsloc hsloc neg def
  frame contents
}def
/frame{
  hsloc mhsloc moveto hsloc hsloc lineto
  mhsloc hsloc lineto mhsloc mhsloc lineto
  closepath stroke

```

```

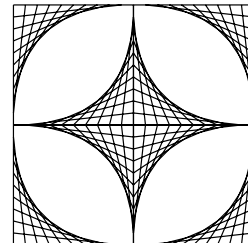
}def
/contents{hsloc 0 moveto
  4{45 rotate .5 hsloc mul 0 lineto
  45 rotate hsloc 0 lineto}repeat
  closepath stroke
}def
/indices[-1.5 -.5 .5 1.5]def
%
hs element .75 setlinewidth
indices{/r exch def
indices{/c exch def
  c abs 1 gt r abs 1 gt or
  {gsave r hs mul c hs mul translate
    .5 hs mul element
    grestore}if
}forall}forall %showpage

```

Explanation. The tile–element–is parameterized over the size. The positioning is prescribed in the array indices. A double loop with an appropriate selection criterion whether a klema should appear. All that matters are the appropriate shifts and the coding of the element. The shifts are the values of the loop variables, apart from a scaling factor. The element is a framed star. The coding of the star has been treated in ‘Stars around I.’ It is rather straightforward, once you look upon it as a problem similar to the drawing of a polygon, namely with a broken line as side.

Example (*2D regular surface*)

This pattern is a remade of my youngest daughter’s hand-work when at primary school. She did only one tile, that is a quarter of the pattern. This example illustrates the advantage of the computer. Once we have the basic element assemblages of translated or rotated copies can be done perfectly and easily.



POSTSCRIPT code by which the picture above is drawn.

```

%!PS-Adobe- Roos' 2D Gabo, cgl Dec 96
%%BoundingBox: -100 -100 100 100
/r 100 def /n 10 def /h r n div def
4{0 1 n{/y exch h mul def
  r y moveto

```

17. As much as a constant like π or e , but lesser known. Maybe if Knuth would adopt this constant...

```

    r y sub r lineto
    y 0 moveto
    0 r y sub lineto
  }for
  90 rotate
}repeat stroke %showpage

```

Remark. Note how the spurious superellipse and cusp emerged as envelopes. A variant code more in the spirit of the earlier given template reads as follows.

```

%!PS-Adobe- Roos' 2D Gabo, cgl Dec 96
%%BoundingBox: -100 -100 100 100
/r 100 def /n 5 def /mn n neg def
/hr .5 r mul def /mhr hr neg def
/h hr n div def
/tile{2{mn 1 n{/y exch h mul def
        hr y moveto y neg hr lineto
      }for 180 rotate
    }repeat
  }def
mhr r hr{/i exch def
mhr r hr{/j exch def
  gsave i j translate
    i j mul 0 lt{90 rotate}if
    tile stroke
  grestore
}for}for %showpage

```

METAFONT code. In this code use has been made of the operator `point` (*expression*) of (*path*).

```

size=50; path p;
p=(origin..controls (0,.6) and (.4,1)..(1,1)
  ..controls (1,.4) and (.6,0)..cycle
  &unitsquare) scaled size;
draw p;
for t:=1 upto 9:
% draw point .1t of p -- point 1+.1t of p;
  draw point 2+.1t of p -- point 3+.1t of p;
  draw point 4+.1t of p -- point 5+.1t of p;
endfor
addto currentpicture also
  currentpicture transformed
  (identity rotated 90 shifted (2size, 0));
addto currentpicture also
  currentpicture transformed
  (identity rotated 180 shifted (2size, 0));
showit; end

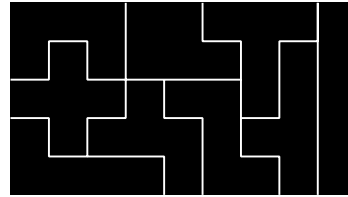
```

Remark. The inner boundary is part of the path because in the real thing the inside 'eye' was also filled with lines. These splines could have been specified equally simple in POSTSCRIPT, but there is as yet not a POSTSCRIPT equivalent of METAFONT's operator `point of`.

2.1 Groups of squares

Combining pentominoes – 5 squares grouped together – is a teaser.¹⁸

Example (*Pentomino puzzle*)



The coding is tedious but straightforward.¹⁹

```

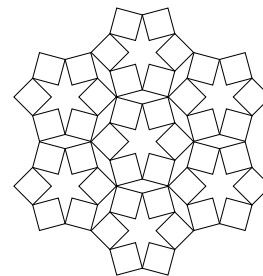
%!PS-Adobe- Pentominoes, cgl Jan 97
%%BoundingBox: -180 -100 180 100
/u 20 def /mu u neg def
/t u 3 mul def /v u 5 mul def
/s u 7 mul def /n u 9 mul def
/mt u -3 mul def /mv u -5 mul def
/ms u -7 mul def /mn u -9 mul def
mn mv moveto n mv lineto n v lineto
  mn v lineto closepath fill
mn u moveto
ms u lineto ms t lineto
mv t lineto mv u lineto
mt u lineto mt v lineto
mt u moveto
%similar for other boundaries
2 setlinejoin 1 setgray
2 setlinewidth stroke %showpage

```

2.2 Non-tight tilings

Non-tight tilings have been exercised by for example Kepler.

Example (*Non-tight tiling of squares*)



¹⁸. Similarly one can group hexagons and so on as pieces for a puzzle.

¹⁹. As with the tangram, when we want to draw more tessellations it is convenient to write definitions for each piece and so on.

Because of the inherent symmetry the coding does not take much lines. Just a (rotated) square as basic element with rotated and translated copies to form the pattern. The diamonds, dodecagons and stars are spurious.

```

%!PS-Adobe- Square madness, cgl Dec 96
%%BoundingBox: -150 -150 150 150
/r 50 def /s r 2.7320 div def
/d r 2 mul s .707 mul sub def
/rotsq{gsave s 0 moveto
  4{0 s lineto 90 rotate}repeat
  closepath stroke grestore
}def
/dodeca{6{gsave r s sub 0 translate rotsq
  grestore 60 rotate}repeat
}def
gsave dodeca grestore
6{gsave 0 d translate dodeca
  grestore 60 rotate}repeat %showpage

```

Explanation. The assembled squares yield spurious dodecagons. I chose as independent parameter for the drawing the radius r of the dodecagon. Its side equals $\frac{r}{2 \sin 75}$. Note that the side of a square equals the side of the dodecagon. The radius s of the circumscribing circle of a square equals $\frac{r}{2\sqrt{2} \sin 75} \approx \frac{r}{2.7320}$. The shift d between dodecagons equals $2r - \frac{s}{\sqrt{2}}$, that is $2r$ minus half of the side, the overlap. Mandatory for the programming are: awareness of the CTM, the use of the graphics state scope delimiters `gsave` and `grestore`, and the difference between user and device space.

METAFONT code.

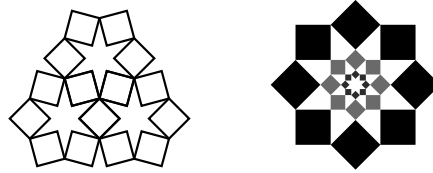
```

%Square madness, cgl Feb 97
r=50; s=r/2.7320; d=2r-.707s;
path square; picture p;
square=((s,0)--(0,s)--(-s,0)--(0,-s)--cycle)
  shifted (r-s,0);
for k=0 step 60 until 300:
  draw square rotated k; endfor
p:=currentpicture;
addto currentpicture also p shifted(0,d);
addto currentpicture also p shifted(0,-d);
addto currentpicture also
  p shifted( .866d, .5d);
addto currentpicture also
  p shifted( .866d,-.5d);
addto currentpicture also
  p shifted(-.866d,-.5d);
addto currentpicture also
  p shifted(-.866d, .5d);
showit; end

```

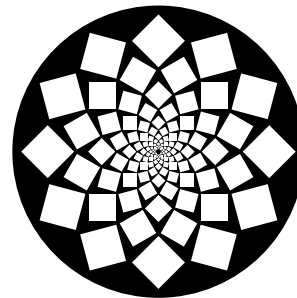
Remark. Note that the picture can be assembled by shifts only. METAFONT does not allow to rotate pictures over arbitrary angles.

The following is left as a finger exercise for the reader.²⁰



Example (Dwirling squares)

Metamorphose or a circle limit? I associate it with the work of Schoonhoven. Well ... what is in the name, who cares.



```

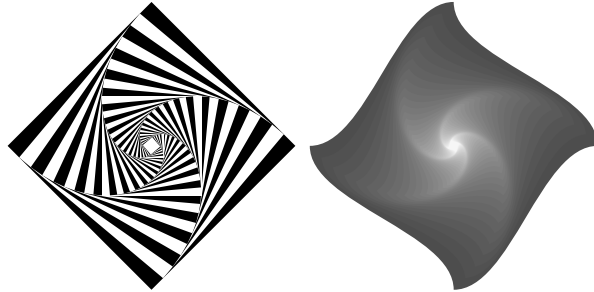
%!PS-Adobe- White squares, cgl Feb 97
%%BoundingBox: -65 -65 65 65
/r 50 def
/hs r 15 sin mul def /s 2 hs mul def
/rotsq{hs 1 sub 0 moveto
  4{90 rotate hs 1 sub 0 lineto}repeat
  closepath fill
}def
/ring{12{gsave r 0 translate rotsq
  grestore 30 rotate}repeat
}def
/rs r hs add 3 add def
/frame{rs 0 moveto 0 0 rs 0 360 arc
  closepath
}def
/f r hs sub r div 15 cos mul def
frame 0 setgray fill
1 setgray
12{ring f f scale 15 rotate}repeat
%showpage

```

²⁰. By the way, it is interesting to see that the band of the right figure is enclosed from the outside by an 8-star and from the inside by a 6-star. However, the drawing can be done simpler.

Explanation. The square replaces a side of a dodecahedron, called ring here. Appropriately rotated and shrunk copies yields the result.

Example (*Shrinking and rotated squares*)
Lauwerier calls this a whirlpool.²¹



The left figure was obtained by the following code.

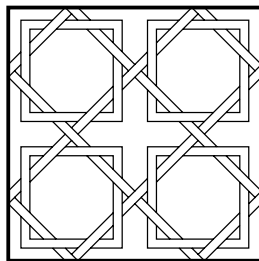
```

%!PS-Adobe- Shrinking squares, cgl Feb 97
%%BoundingBox: -100 -100 100 100
/r 100 def /alpha 5 def
/c 1 alpha cos alpha sin add div def
/square{r 0 moveto
  0 r lineto r neg 0 lineto 0 r neg lineto
  closepath sof}def
/sof{fill}def /flipflop true def
36{flipflop{0 setgray}{1 setgray}ifelse
  square
  /flipflop flipflop not def
  /r r c mul def
  alpha rotate}repeat %showpage
  
```

Explanation. At the heart is a square parameterized over r . An appropriate use of setgray during rotation and shrinking yields the result.

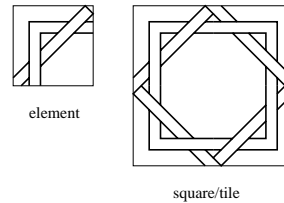
2.3 Classical tilings

Intriguing classical tilings are composed of simple geometrical patterns of which lines continue beyond the boundaries of the basic element, to form plaits.



clipped pattern

How to draw these? What is to be considered as the basic element?



As can be seen from the above, I choose a hook together with an overlaced strip as basic element. This element is not symmetric. A tile is composed of 4 of these elements rotated over $90k^\circ$, for $k = 0, 1, 2, 3$, respectively. The resulting tile is (rotational, 4-valent) symmetric. Combining (translated) copies yields the pattern. The POSTSCRIPT code for the pattern above reads as follows, apart from the labeling.

```

%!PS-Adobe- Squares broidery, cgl Dec 96
%%BoundingBox: -100 -120 100 100
/s 50 def /ms s neg def
/d 5 def /dst d 1.414 mul def
/md d neg def
/a s d 2 mul sub def /ma a neg def
/aa a dst sub def /maa aa neg def
/element{ms d moveto
  ma d 3 mul lineto
  maa d 3 mul dst add moveto
  md s lineto
  0 s d sub lineto
  maa d dst add lineto
  ma d moveto
  ma d sub 0 lineto
}
%
ma 0 moveto ma a lineto d -3 mul a lineto
md a moveto 0 a lineto
maa 0 moveto maa aa lineto
d -3 mul dst sub aa lineto
md dst sub aa moveto 0 aa lineto stroke
}def
%
/tile{4{element 90 rotate}repeat}def
%
gsave ms s translate tile grestore
gsave s s translate tile grestore
gsave ms ms translate tile grestore
gsave s ms translate tile grestore
  
```

21. He treats the general problem of rotating and shrinking of a polygon in general. H.A Lauwerier(1987): Meetkunde met de microcomputer. Epsilon 8. Jackowski – see EuroT_EX 95 proceedings – used splines as sides and obtained beautiful results. The (right) figure is in his spirit. When I had nearly finished this paper another booklet by Lauwerier ‘Symmetrie etc.’ came my way. In that he explains Escher’s approach similarly as I did. An eye-opener was his mentioning of the art of Hinterreiter, where metamorphoses are seen as a form of perspective. So just a transformation of the regular forms, that simple.

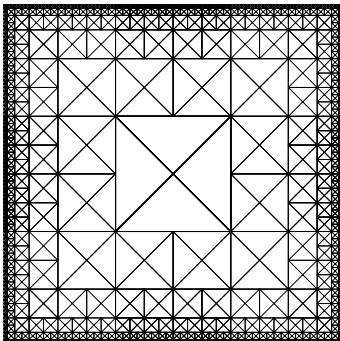
```
%frame
/ts s 2 mul def /fs ts ts add def
ts neg dup moveto fs 0 rlineto
0 fs rlineto fs neg 0 rlineto closepath
3 setlinewidth stroke %showpage
```

The coding in METAFONT goes similarly, because pictures can be rotated over $90k^\circ$, for $k = 1, 2, \dots$

2.4 Square limits

Michel Goossens in his ‘ \LaTeX en PostScript – de complementariteit in de praktijk’ has shown an example of this class of Eschers. Lengthy PostScript code borrowed from the net. The picture suffers from too much blackness near the boundary, IMHO, with all respect. This is an interesting aspect that we should draw thinner curves when they accumulate, because of the optical effect.

Example (Square limits grid)



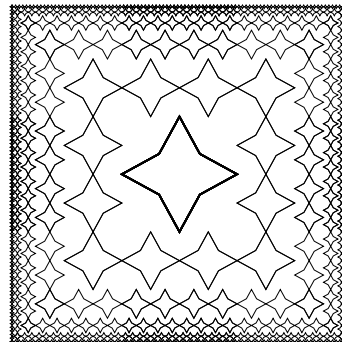
The code below is interesting because it demonstrates the use of recursion in POSTSCRIPT together with the use of the operand stack. Note that the line thickness also diminishes and that drawing is stopped overall at a certain depth, which I could not achieve via `scale`. Also interesting is the use of reflection about the line $y = x$. The code can be used as template for tiling – in the sense of Escher’s limit process – with scaled (and rotated) copies of a master tile created with the origin as center. Here the (contents of the) tile – element – is simply a cross.²²

```
!PS-Adobe Square limits grid, cgl Feb 97
%%BoundingBox: -96 -96 96 96
/gs 64 def /mgs gs neg def
/square{%lw s x y on stack
/y exch def /lx exch def
/hs exch 2 div def /mhs hs neg def
/llw exch def
gsave lx y translate llw setlinewidth
element
grestore
.5 llw mul hs lx 1.5 hs mul add
```

```
y .5 hs mul add
.5 llw mul hs lx 1.5 hs mul add
y .5 hs mul sub
hs 1 gt {square square}
{8{pop}repeat} ifelse
}def
/element{hs mhs moveto hs hs lineto
mhs hs lineto mhs mhs lineto
closepath stroke
hs mhs moveto mhs hs lineto
mhs mhs moveto hs hs lineto
.25 llw mul setlinewidth stroke
}def
%
/pattern{4{/lw 1 def
/s gs def /ms s neg def /x 0 def
gsave
6{gsave
2{lw s x x square
-1 1 scale 90 rotate
}repeat
grestore
/s .5 s mul def /lw .5 lw mul def
/x x 1.5 s mul add def
}repeat
grestore -90 rotate
}repeat
}def
%
pattern %showpage
```

Escher enriched this grid with fishes for example. Maybe I’ll redo that example in due time.²³

Example (Variant with stars)

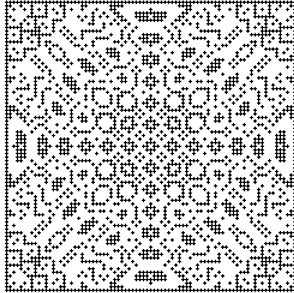


22. The Sierpiński gasket – a fractal with limits all over – will be treated in the note on fractals.

23. On the net a larger code is available, of which the result is incorporated in Goossens’ article.

Example (*Broideries*)

Lauwerier introduced broideries. The tile is defined by a function with inherent symmetries. Lauwerier used among other things reflection symmetrie around the x -axis, y -axis, and the lines $y = \pm x$, for example as in the function $f(x, y) = (1 - x^2)(1 - y^2)$. The idea of broideries are obtained if at a point (x, y) a mark is drawn when the function value satisfies a certain criterion. In the broiderie below a mark—small cross—is drawn at (x, y) if $\text{int}(400 f(x, y))$, equals a multiple of 3, for $x = -1, \dots, -1/n, 0, 1/n, \dots, 1, y = -1, \dots, -1/n, 0, 1/n, \dots, 1, n = 40$ was used.

**3 Triangles**

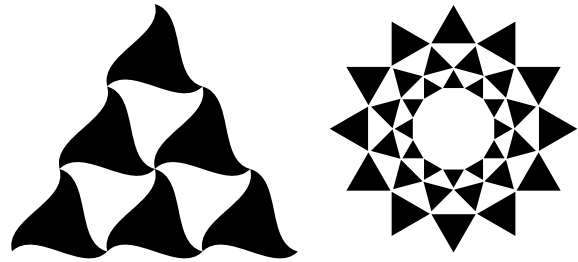
Triangles tile well, especially in space. For example they are used as elements in approximating surfaces among other things in analogy with line pieces for approximating curves in the plane. They form the sides of the Platonian solids tetraeder, octaeder, and icasohedron. I'll restrict myself to triangles in the (xy) -plane.

Example (*Puzzle of groups of triangles*)

As with tangrams we can write definitions for the pieces and so on.

**Example** (*Flexed triangles and coronas of triangles*)

We can modify the sides or arrange them in patterns.



The codes for these drawings read as follows.

```

%!PS-Adobe- Flexed triangles, cgl Feb 97
%%BoundingBox: -15 -15 115 115
/s 50 def
/hs .5 s mul def /mhs hs neg def
/hss 1.732 hs mul def
/qs .5 hs mul def /mqs qs neg def
/r 1.15 hs mul def
/hr .5 r mul def /mhr hr neg def
/tri{mhs mhr moveto
    3{mqs mhr qs add
        qs mhr mqs add hs mhr curveto
        120 rotate
    }repeat fill
}def
tri
gsave 2{s 0 translate tri}repeat grestore
gsave hs hss translate tri
    s 0 translate tri
grestore
s 2 hss mul translate tri %showpage

```

I did not rotate to allow more easily extension of the pattern.

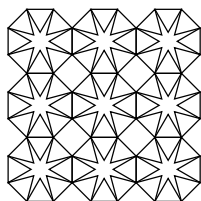
```

%!PS-Adobe- Coronas of triangles, cgl Feb 97
%%BoundingBox: -72.5 -72.5 72.5 72.5
/r 50 def %BB: r(1+1.732 sin 15)
/hs r 15 sin mul def /mhs hs neg def
/hss 1.732 hs mul def
/f r r hss add div def
/tri{hss 0 moveto
    0 hs lineto 0 mhs lineto
    closepath fill
}def
3{12{gsave r 0 translate tri grestore
    30 rotate
    }repeat f f scale 15 rotate
}repeat %showpage

```

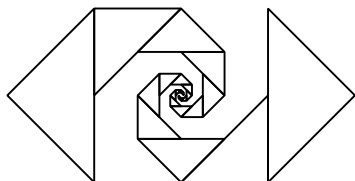
Note the use of scale.

Example *((Non-tight) triangles)*



For the coding the same template as for the square tiles was used. Note the spurious squares, stars and octagons.

Example *(Triangular arms: east-west rencontre)*



With the following (tail) recursion code.

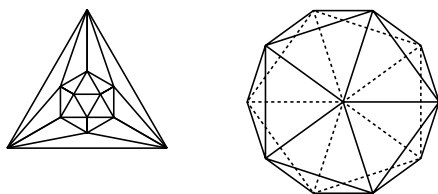
```

%!PS-Adobe- Triangle arms, cgl Feb 97
%%BoundingBox: 0 -50 185 50
/s 64 def /ts 2 s mul def
/tri{/locs exch def
      0 locs moveto locs 0 lineto 0 0 lineto
      closepath currentpoint stroke
      translate -45 rotate
      /locs .707 locs mul def
      locs 1 ge {locs tri} if
    }def
2 setlinejoin -45 rotate
gsave s tri grestore
ts ts translate 180 rotate
s tri %showpage

```

Example *(Graph of icosahedron)*

The icosahedron has triangles as sides. The following projection shows them all. ‘Tiling’ to assist insight.



The left projection is obtained as follows.

```

%!PS-Adobe- Graph icosaheder, cgl Mrt 97
%%BoundingBox: -50 -30 50 60
/s 100 def /ss 1.732 s mul def

```

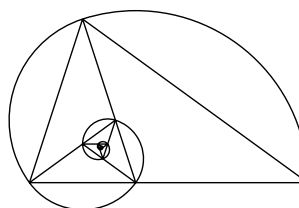
```

2 setlinejoin
/p1{.1667 s mul -.0555 ss mul}def
/p2{.5 s mul -.1667 ss mul}def
/p3{0 -.1111 ss mul}def
/p4{0 -.0555 ss mul}def
3{gsave
  2{p1 moveto p2 lineto p3 lineto
    p4 lineto p1 lineto p3 lineto
    -1 1 scale}repeat stroke
  grestore 120 rotate}repeat
p2 moveto
3{120 rotate p2 lineto}repeat stroke
p4 moveto
3{120 rotate p4 lineto}repeat stroke
%showpage

```

The right projection is composed of pentagons and a decahedron.

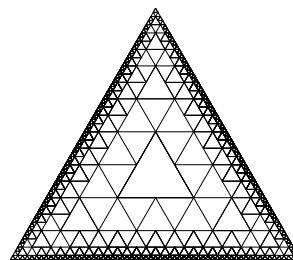
Example *(Nested triangles)*



This triangle comes back in pentagons: a side with two diagonals. The golden ratio is in there, and we can split of similar smaller triangles infinitely. The code is in the same spirit as that for rectangles.

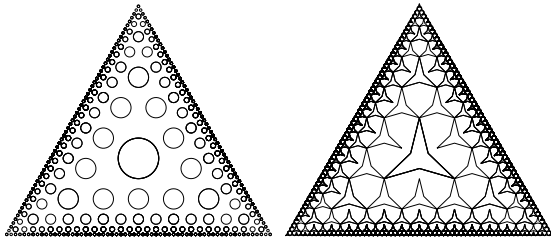
Example *(Triangular limits grid)*

In analogy with the square limits grid there is the triangular limits grid, which is related to the Sierpiński carpets, let us say it is an outbound variant. I did not find this one in Escher’s work. Maybe too much complexity in the corners?

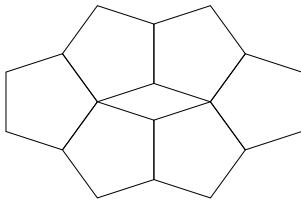


The code is in the same spirit as the code for the limit squares.²⁴

24. The Sierpiński gasket – a fractal which is limiting allover – will be treated in the note on fractals.

Example (*Variants*)**4 Pentagons**

Pentagons don't tile tight. So what?

Example (*(Non-tight) pentagons I; lips*)

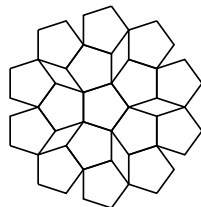
In the coding the reflection via scale is interesting.

```

%!PS-Adobe- Simple Pentas, cgl Feb 97
%%BoundingBox: -175 -95 175 95
/r 50 def /mr r neg def
/rin r 36 cos mul def
/mhdx s 18 cos mul neg def
/pentagon{r 0 moveto
  5{72 rotate r 0 lineto}repeat
  stroke
}def
%
/figure{pentagon}def
2{gsave mhdx mr add 0 translate
  figure
  2{gsave 36 rotate 2 rin mul 0 translate
    figure
    grestore 1 -1 scale
  }repeat
  grestore -1 1 scale
}repeat %showpage

```

Dürer tiled straightforwardly as follows. Only spurious diamonds show up.

**Example** (*(Non-tight) pentagons II*)

Because of the inherent symmetry the coding does not take much lines. Just a pentagon as basic element with rotated and translated copies. The diamonds are spurious. Do you see how to code another ring, how to let the pattern grow?

```

%!PS-Adobe- Duerer, cgl Jan 97
%%BoundingBox: -55 -65 55 65
/r 15 def /rin r 54 sin mul def
/p{gsave r 0 moveto
  5{72 rotate r 0 lineto}repeat
  stroke grestore
}def
/tr{2 rin mul 0 translate}def
p 36 rotate
5{gsave tr
  p gsave 36 rotate tr p grestore
  gsave -36 rotate tr p grestore
  grestore 72 rotate}repeat %showpage

```

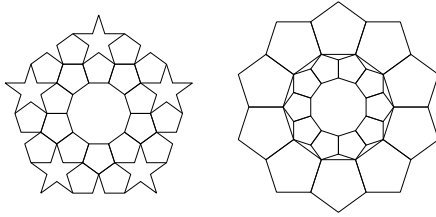
METAFONT code. My earlier METAFONT code did *not* take into account the fact that the diamonds are spurious. Part of it is given below. Remarkable is that use is made of `reflectedabout` which is not needed in the simpler approach as worked out in the later POSTSCRIPT code.

```

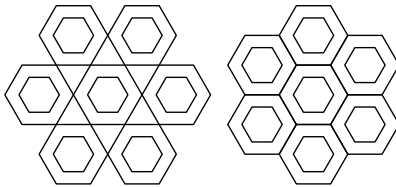
r=15; n=5; s=2r*sind36; path p[];
diag=s*(1+2sind18);
height=s*(cosd54+cosd18);

p1=(r,0)--(r*cosd72, r*sind72);
p2=(r,0)--(r+s*cosd18,s*sind18)
  --(r+s*cosd18,s*(1+sind18))
  --(r,diag)--(r*cosd72, r*sind72);
p3=(r+s*cosd18,s*sind18)
  --(r+2s*cosd18,0)--
  (r+s*cosd18,-s*sind18);
p41=(r+2s*cosd18,0)
  --(r+2s*cosd18+s*cosd54,s*sind54)
  --(r+2s*cosd18,diag)
  --(r+s*cosd18,s*(1+sind18));
p42=p41 reflectedabout
  (origin,(r+s*cosd18,s*(1+sind18)));
for k=72step72until360:
  draw p1 rotated k;endfor
for k=0step72until360:
  draw p2 rotated k;endfor
for k=0step72until360:
  draw p3 rotated k;endfor
for k=0step72until360:
  draw p41 rotated k;draw p42 rotated k;
endfor
showit; end

```

Example (*Decahedron at the heart*)**5 Hexagons**

Hexagons tile tight and may yield intriguing patterns. Notice the difference in the optical effect.

Example (*Hexagon template tilings*)

POSTSCRIPT code. Hexagon tiling templates.

```

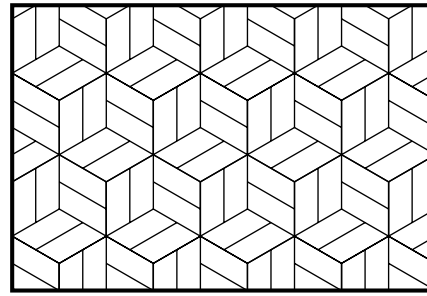
%!PS-Adobe 6-stars, cgl Feb 97
%%BoundingBox: -75 -70 220 70
/r 25 def 2 setlinejoin
/sixtile{/rloc exch def %r on stack
  rloc 0 moveto
  6{60 rotate rloc 0 lineto}repeat
  closepath stroke
}def
%tile at corners
r sixtile .588 r mul sixtile
6{gsave 2 r mul 0 translate
  r sixtile .588 r mul sixtile
  grestore
  60 rotate}repeat
%tile side by side
6 r mul 0 translate
r sixtile .588 r mul sixtile
30 rotate
6{gsave 1.732 r mul 0 translate 30 rotate
  r sixtile .588 r mul sixtile
  grestore 60 rotate}repeat %showpage

```

Remark. This template will be reused when tiling hexagrams later on.

Example (*Horak's cubes*)

An optical effect is obtained, it looks like projected cubes. The minimal information – the parameter of the figure – is the radius r of the circumscribing circle of the hexagon.



POSTSCRIPT code. A hexagon is underneath, that is a cube in projection. Starting from this the code is straightforward.

```

%!PS-Adobe-Horak's tiles, cgl Jan 97
%%BoundingBox: 0 0 310 205
/r 40 def
/s r 2 div def /ms s neg def
/sst s 1.732 mul def
/ts s 2 mul def /mts ts neg def
/hs s 2 div def
/dia{0 0 moveto 0 ts lineto
  sst ms rlineto 0 mts rlineto
  0 s rmoveto 0 s lineto
}def
/tile{gsave 3{dia 120 rotate}repeat
  stroke grestore
}def
/pattern{gsave
2{gsave
  5{gsave tile sst s 3 mul translate
    tile
    grestore sst 2 mul 0 translate
  }repeat
  grestore 0 s 6 mul translate
}repeat grestore
}def
/a s 10.5 mul def /ma a neg def
/frame{0 0 moveto
  0 a rlineto s 15.5 mul 0 rlineto
  0 ma rlineto closepath
}def
gsave frame clip pattern grestore
frame 3 setlinewidth stroke %showpage

```

Explanation. The radius r of the hexagon is $2s$. An incomplete diamond – side of a cube – is drawn and rotated over 120° and 240° to form the basic hexagon tile, with the origin at its centre. The pattern emerges by copying hexagons translated over $k(s\sqrt{3}, 3s)$, and $k(2s\sqrt{3}, 0)$, for $k = 1, 2, \dots$

It hints at how to draw 3D figures: draw the projection.²⁵

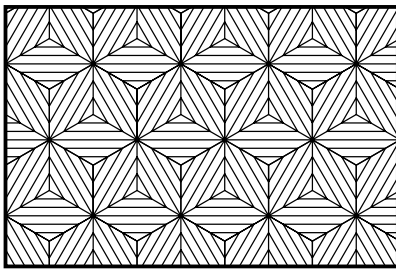
25. This does not mean that we should specify the projected figure in general. I prefer to specify the figure in 3D and use one of my

METAFONT code.

```
s=40;path p[]; %cgl Jan 97
p1=origin--(0,2s);
p2=p1 rotated-120 shifted(0,s);
p3=p1 & p2 shifted(0,s);
p4=p1 shifted(1.732s,-s);
for k=0 step-120 until-240: %draw tile
  draw p2 rotated k;
  draw p3 rotated k;
  draw p4 rotated k;endfor
for k=1upto2: %draw pattern
  addto currentpicture also currentpicture
    shifted (k*(1.732s,3s));
  addto currentpicture also currentpicture
    shifted (3.464s*k,0);
endfor
%Cut out a piece
cullit;
fill unitsquare xscaled 10size
  yscaled 7.5size shifted (2size,0);
cull currentpicture keeping (2,infinity);
%Frame the piece
pickup pencircle scaled 3;
draw unitsquare xscaled 10size
  yscaled 7.5size shifted (2size,0);
showit; end
```

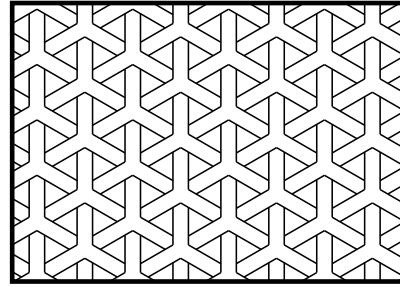
Remarks. The clipping must be done differently in METAFONT, because the concept of a clipping boundary is not there. POSTSCRIPT allows any path. METAFONT requires as path a continuous line. As a consequence I have split up the basic element, although I could have traversed various parts of the element more than once.

Example (*Variant*)



Example (*Romanovsky's Chinese porcelain*)

A broderie of overlapping hexagons. The minimal information required to draw this pattern is the height s and the width w of the basic bar.



POSTSCRIPT code.

```
%!PS-Adobe-
% Romanovsky's porcelain, cgl Jan 97
%%BoundingBox: 0 0 255 180
/s 30 def /ms s neg def /ss s 1.732 mul def
  /hs s 2 div def /mhs hs neg def
  /hss hs 1.732 mul def
/w s 3 div def /wds w 1.732 div def
  /hw w 2 div def
  /hwds hw 1.732 div def
/line{hw s hwds sub moveto hw hwds lineto
}def
/tile{3{gsave line -1 1 scale line stroke
  grestore 120 rotate
}repeat
}def
/pattern{gsave
6{gsave
  7{gsave tile
    hw hss add hs hwds add translate tile
    grestore w ss add 0 translate
  }repeat
  grestore 0 s wds add translate
}repeat grestore
}def
/a s 6 mul def /ma a neg def
/frame{0 0 moveto 0 a rlineto
  s 8.5 mul 0 rlineto 0 ma rlineto
  closepath
}def
gsave frame clip newpath
  mhs mhs translate pattern
grestore
frame 3 setlinewidth stroke %showpage
```

Explanation. As height I took the radius of the circumscribing hexagon. The tile is a hexagon, with 6 inner line pieces. The basic element is drawn by exploiting fully its symmetry. Once the shifts have been calculated the coding is straightforward. The pattern results after first adding a copy translated by $(s\sqrt{3} + w, s + w/\sqrt{3})/2$, and then tile

pointtopair projection operators.

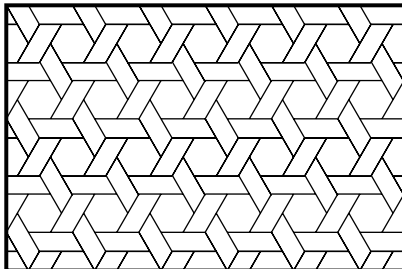
this with horizontal and vertical translations $k(s\sqrt{3}+w, 0)$, and $k(0, s + w/\sqrt{3})$, for $k = 1, 2, \dots$

METAFONT code. Note that in METAFONT we can't rotate pictures arbitrarily, but we can do with paths. The code below I wrote a year earlier than the POSTSCRIPT code, and differs a little, if not for the basic element.

```
s=30; w=.333s; path p[]; %cgl, Jan 97
p1=(.5w, .288w)--(.5w,s-.288w);
p2=(-.5w, .288w)--(-.5w,s-.288w);
for k= 0 step 120 until 240: %tile
  draw p1 rotated k; draw p2 rotated k;
endfor
addto currentpicture also currentpicture
  shifted(.866s+.5w, .5s+.2887w);
for k= 1 step 1 until 2:
addto currentpicture also currentpicture
  shifted(k*(0, s+.5774w));
addto currentpicture also currentpicture
  shifted(k*(1.732s+w,0));
endfor
cullit;%clipping boundary functionality
fill unitsquare xscaled 6.5s
  yscaled 4s shifted(.25s,0);
cull currentpicture keeping (2,infinity);
pickup pencircle scaled 3; %frame
draw unitsquare xscaled 6.5s
  yscaled 4s shifted(.25s,0);
showit; end
```

Example (Variant hex pattern)

And what about the following with only the side of the hexagon as parameter?



6 Escher

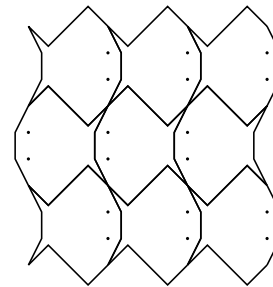
M C Escher, a Dutch (graphics) artist of the first half of the XX-th century, has among other things enriched classical tilings by loosening the strict geometrical tradition. He drew creatures like Buddhas, reptiles and so on, appealing to the masses, well ... science biased.

Similar to the above classical plait his basic element is not symmetric.²⁶

He realized that the picture on the tile is only restricted by the points where it cuts the boundary. Maybe he just thought of deforming the boundaries. Moreover, the boundaries are related by symmetry, induced by tiling.

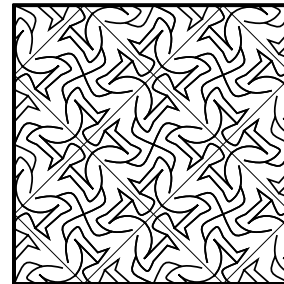
Example (Escher-like fishes)

The following is borrowed from Lauwerier's earlier mentioned booklet 'Symmetrie etc.' The tiling is much in the spirit of our earlier template. The basic tile consists of 2 lines, one also translated and the other also reflected. The middle row consists of reflected tiles: fishes swimming in the other direction.



6.1 Escher's Squares

Example (Escher's Buddhas)



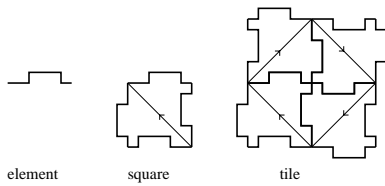
How to draw these? How to program?

Analyzing the picture reveals that the basic element is a modified side of the square. This side can be rotated and reflected to yield a square. Rotated copies of this square form a tile. Translated copies will yield the pattern.

Below the basic element has been simplified into a line with a notch, and the intermediate phases have been shown.

²⁶ MacGillavry, a Dutch crystallographer, has studied Escher's symmetries and published on the issue.

Example (*Escher's mechanism*)



Programming the four phases

$$element \xrightarrow{R,M} square \xrightarrow{R} tile \xrightarrow{T} pattern$$

can be read from the following POSTSCRIPT code for Escher's Buddhas. (*R, M, T* denote *Rotation, Mirroring* and *Translation*, respectively.)

```

%!PS-Adobe- Escher Buddhas, cgl Dec 96
%%BoundingBox: -20 -45 320 140
/s 20 def /ms s neg def /ts s s add def
/pa{s s}def
/pb{.7 s mul 1.7 s mul}def
/finger{.2 s mul 1.5 s mul}def
/pc{-.4 s mul 1.45 s mul}def
/pd{.5 s mul 1.35 s mul}def
/knee{.7 s mul 1.15 s mul}def
/pf{.35 s mul s}def
/ankle{-.1 s mul .6 s mul}def
/toe{0 .3 s mul}def
/heel{-.4 s mul .6 s mul}def
/head{-.9 s mul .9 s mul}def
/center{-.25 s mul 1.8 s mul}def
/cpa{0 1.55 s mul}def
/cpb{.25 s mul 1.55 s mul}def
/cpc{-.6 s mul 1.1 s mul}def
%
/element{gsave
pa moveto pb pb finger curveto
pc moveto cpa cpb pd curveto
knee knee pf curveto %pf lineto
ankle lineto
toe lineto
heel lineto
center .75 s mul 315 225 arcn
heel moveto cpc cpc head curveto
stroke grestore}def
%
/tile{element
gsave 90 rotate 0 s -2 mul translate
element
grestore gsave 1 -1 scale 90 rotate
element
grestore gsave -1 1 scale

```

```

0 s -2 mul translate
element
grestore gsave ms s moveto s ms lineto
.05 setlinewidth stroke grestore
}def
%
/pattern{gsave
tile s 4 mul 0 translate
tile 0 s 4 mul translate
tile s -4 mul 0 translate
tile grestore
}def
%
/as 8 s mul def /mas as neg def
/contour{ms ms moveto
as 0 rlineto 0 as rlineto
mas 0 rlineto closepath
}def
%
/Times-Roman findfont
10 scalefont setfont

%Draw
element
ms ms 25 sub moveto (element)show
65 0 translate square
ms ms 20 sub moveto (square)show
75 0 translate tile
0 ms 20 sub moveto (tile)show
110 0 translate
0 ms 20 sub moveto (clipped pattern)show
contour clip pattern
contour 2 setlinewidth stroke %showpage

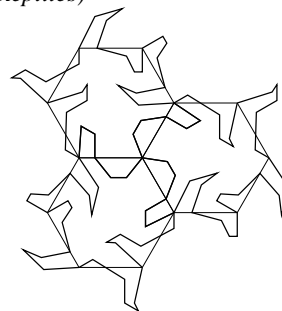
```

My METAFONT code is similar to the code for the reptiles given below.

6.2 Escher's hexagons

The same mechanism as demonstrated with squares was applied by Escher to a hexagonal grid.

Example (*Reptiles*)



METAFONT code.²⁷

```

%Escher's Reptiles, cgl May 96
pickup pencircle scaled 1;
pair p[]; path ctoad;
s:=50;
%hexagon p1--p3--p6--p8--p10--p12--cycle
%as grid/canvas
p3=s*up;
p6=p3 rotated60;
p8=p6 rotated60;
p10=p8 rotated60;
p12=p10 rotated60;
p1=p12 rotated60;
%pickup pencircle scaled .05;
%draw p1--p3--p6--p8--p10--p12--cycle;
p2=.3333[p3,p1];p13=.3333[p12,p1];
p4=.3333[p3,p6];p7=.3333[p8,p6];
p5=.3333[p6,p3];
p9=.3333[p10,p8];
p11=.3333[p10,p12];
%pickup pencircle scaled 5;
%for k=1 upto 13: drawdot p[k];endfor
p14=.16667[p1,p6] + .1s*down;
p141=p14 +1.1s*(-.17, .1);
p15=.25[p3,p10];
p16=.16667s*(-1,1);
p17=p16 + .3333s*up;
p18=p5 + .16673s*(.2, -1);
p19=p18 - s*(.1, .1);
p20=p19 +.3333s*down;
p21=p8 + s*(.28,0);
p22=.22[p9,p2];
p23=.3[p11,p4];
p24=p12 + 1.5s*(-.17, .1);
p25=p24-.1s*(1, .5);
p26=p25+.1s*left;
p27=p26+.3333s*up;
p28=.2[p13,p6];
pickup pencircle scaled 2;
%a complete toad, well reptile
ctoad=p1--(p1--p14--p141--p2)
rotatedaround(p1,120)--
p13--p28--p27--p26--p25--p24--p12--
(p8--p21--p9)rotatedaround(p1,-120)
shifted (0,-3s)--
p11--p23--p22--p10--
(p10--p22--p23--p11)
rotatedaround(p10,120)--
p9--p21--p8--
(p3--p15--p16--p17--p4)
rotatedaround(p1,-120)
shifted (-1.5s*sqrt3,-1.5s)--

```

```

p7--p20--p19--p18--p5--p6--
(p6--p5--p18--p19--p20--p7)
rotatedaround(p6,120)--
p4--p17--p16--p15--p3--
(p12--p24--p25--p26--p27--p28--p13)
rotatedaround(p1,-120)--
p2--p141--p14--p1;
draw ctoad;
draw ctoad rotatedabout (p1,120);
draw ctoad rotatedabout (p1,-120);
showit; end

```

Explanation. This code is based on points on the circumference of the hexagon. These are related by symmetry induced by the tiling as can be seen from the figure. A more extensive pattern can be obtained by adding shifts to the current picture.

POSTSCRIPT code by which the included figures are drawn.

```

%!PS-Adobe- Escher's reptiles, cgl Jan 97
%BoundingBox: -50 -70 130 150
/r 50 def /mr r neg def
/hr r 2 div def /mhr hr neg def
/qr r 4 div def /mqr qr neg def
/er r 8 div def /mer er neg def
/hrx hr 1.732 mul def /mhrx hrx neg def
/grx hrx 2 div def /mqrq grx neg def
/erx grx 2 div def /merx erx neg def
/delta er 2 div def
/head{hr 0 moveto
qr mqr 1.25 mul lineto
0 mqr 1.25 mul lineto
mqr 0 lineto
mqr qr 1.5 mul lineto
merx er rlineto
mhr 0 lineto
stroke
}def
/side{hr 0 moveto
mer 2 div delta sub
mhr delta sub rlineto
delta mer rlineto
merx er rlineto
0 hr er sub rlineto
mhr erx add er lineto
mhr 0 lineto stroke
}def
/tail{hr 0 moveto
qr mhr lineto
mqrq mer rlineto %end

```

27. A tedious code. An exercise in MF programming. My late POSTSCRIPT code is simpler.

```

qr er 1.5 mul rlineto
erx 0 lineto
mqr er lineto
mer qr rlineto
mer 0 rlineto
mhr 0 lineto stroke
}def
/reptile{
gsave 0 hrx translate head grestore
gsave hr qr add qrx translate
120 rotate head grestore
gsave 0 mhrx translate
side grestore
gsave hr qr add mqr translate
-120 rotate side grestore
gsave mhr mqr add qrx translate
60 rotate tail grestore
gsave mhr mqr add mqr translate
-60 rotate tail grestore
}def
reptile
/grid{gsave r 0 moveto
6{60 rotate r 0 lineto}repeat
.05 setlinewidth stroke grestore
}def
grid
gsave r hr add hr translate
120 rotate grid reptile grestore
gsave 0 hrx 2 mul translate
-120 rotate grid reptile grestore
%showpage

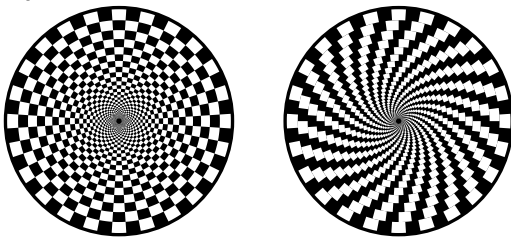
```

Explanation. This code is based on the symmetry of pairs of sides of the hexagon: head, side and tail. Open as yet is how to compose contours from these for coloring purposes.

6.3 Circles

With circles we have two special locations: the centre and the boundary. I associate this with implosion and explosion. When convergence is towards the centre it is about a centre of multiplication. When convergence is outbound I'll talk about a circle limit, in pursuit of Escher.

Example (Disk divisions)

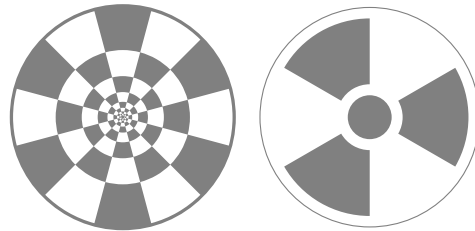


```

%!PS-Adobe- Circular checker board, cgl Feb 97
%%BoundingBox: -100 -103 350 103
/r 100 def
/sq{rin 3.75 cos mul rin -3.75 sin mul moveto
0 0 rout -3.75 3.75 arc
0 0 rin 3.75 -3.75 arcn fill
}def
/ring{/rin rout 1 1.5 3.75 sin mul sub mul def
24{gsave sq grestore 15 rotate
}repeat /rout rin def
}def
/frame{rf 0 moveto 0 0 rf 0 360 arc}def
%
/rout r def /rf rout 3 add def
gsave frame 0 setgray fill 1 setgray
36{ring 7.5 rotate}repeat
grestore
250 0 translate %spirals
/rout r def /rf rout 3 add def
frame 0 setgray fill
gsave 1 setgray 36{ring 3.75 rotate}repeat
grestore %showpage

```

Example (Darts board and radioactive radiation logo)



```

%!PS-Adobe- Circular checkerboard, cgl Feb 97
%%BoundingBox: -105 -105 330 105
/r 100 def
/sector{rin 15 cos mul rin -15 sin mul moveto
0 0 rout -15 15 arc
0 0 rin 15 -15 arcn
closepath fill
}def
/ring{/rin rout 1 1.5 15 sin mul sub mul def
6{gsave sector grestore 60 rotate
}repeat /rout rin def
}def
/frame{rf 0 moveto
0 0 rf 0 360 arc
closepath
}def
%
gsave
/rout r def
/rf rout 3 add def

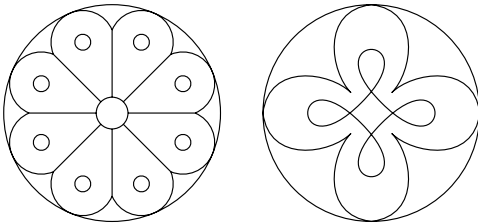
```

```

frame 0.5 setgray fill
1 setgray
9{ring 30 rotate}repeat
grestore
%Radioactive radiation logo
225 0 translate
/rout r 10 sub def /rin rout 3 div def
/rf r def
frame .5 setgray stroke
/sector{rin 30 cos mul
      rin -30 sin mul moveto
      0 0 rout -30 30 arc
      0 0 rin 30 -30 arcn
      closepath .5 setgray fill
}def
3{gsave sector grestore 120 rotate}repeat
rin 10 sub 0 moveto
0 0 rin 10 sub 0 360 arc .5 setgray fill
%showpage

```

Example (*Windows: From a barn and from Malbork*)

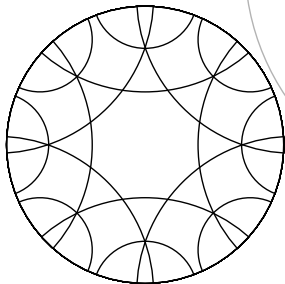


6.4 Circle limits

Escher also used hyperbolic grids: circular arcs within a circle, where the centers of the circles converge towards the boundary, and the arcs cut the boundary perpendicularly.

It did take me some time to realize what is really meant by circle limits. The basics, unblurred by Escher's approach, is really simple, and in analogy with triangular and square limits, and ... as it turned out easier to code as well. Let us first do it à la Escher followed by a straight one.

Example (*Circle limits grid à la Escher*)



```

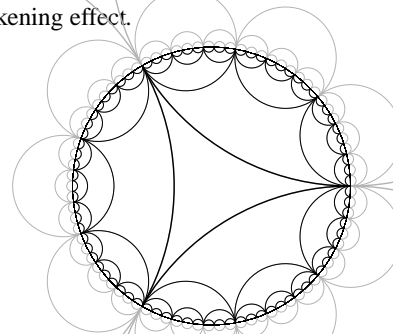
%!PS-Adobe- Circle limits, cgl Jan 97
%%BoundingBox: -100 -100 100 100
/R 100 def /tR 2 R mul def
/a .38 R mul def
/m .5 R R a div mul a add mul def
/r m a sub def /tr 2 r mul def
/circle{%x y r on stack
      /rad exch def
      /y exch def /x exch def
      x y translate
      rad 0 moveto
      0 0 rad 0 360 arc
}def
0 0 R circle clip
%
2{8{gsave m 0 r circle stroke grestore
      45 rotate}repeat
      /r r 3 div def
      /m R R mul r r mul add sqrt def
}repeat
0 0 R circle 2 setlinewidth stroke %showpage

```

Escher extended this also to the surface of a sphere, for example in his 'Angels and Devils.'

Example (*The straight circle limits grid*)

In analogy with squares and triangles the straight circle limits are even easier to code. The starting point is the starting number of arcs n as parameter. 360 divided by n yields the arc of the main circle cutout by the biggest circle of the limit set of circles. The centre and radius of the circle follow easily from the data of the main circle, the cutout arc, and that the circles intersect orthogonally. Rotate the template – ring – and loop this `levels` times. Note that en-passant the line thickness is decreased to counteract the blackening effect.



```

%!PS-Adobe- Straight circle limits, cgl Feb 97
%%BoundingBox: -100 -100 100 100
/R 100 def /levels 4 def
/circle{%x y r on stack
      /radius exch def /y exch def /x exch def
      x y translate

```

```

radius 0 moveto 0 0 radius 0 360 arc
}def
/ring{/n exch def %n on stack
/ang 360 n div def
/r R .5 ang mul sin
.5 ang mul cos div mul def
/m R R mul r r mul add sqrt def
gsave -.5 ang mul rotate
n{gsave m 0 r circle stroke grestore
ang rotate}repeat
grestore
}def
/picture{/n 3 def /f n def /wl 1 def
levels{wl setlinewidth
n ring /n f n mul def
/wl .5 wl mul def
}repeat
}def
gsave .925 setgray picture grestore
0 0 R circle clip picture 0 0 R circle
1 setlinewidth stroke %showpage

```

Circles within a circle Apollonius already considered circles which touch on 3 circles.

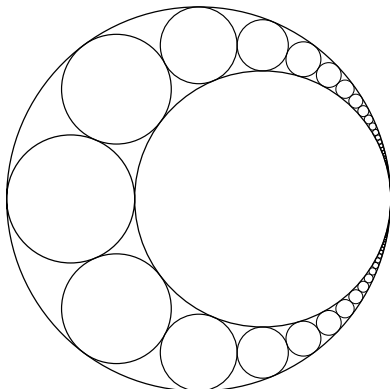
The equations for a circle, in terms of its centre and radius $\{x, y, \delta\}$, which touches on 3 other circles are

$$(x - x_i)^2 + (y - y_i)^2 = (r_i \pm \delta)^2, i = 1, 2, 3$$

with $\{x_i, y_i, r_i\}_{i=1}^3$ the data of the given circles. \pm accounts for touching on the outside, respectively inside.

For the covering below the 3 quadratic equations can be simplified into 2 linear equations and a quadratic equation. These linear equations have been made explicit as function of δ for the centres of circles which touch the main circle and the circumscribing circle.

Example (Circle covered by circles)



Explanation. I chose as origin the midpoint of the circumscribing circle. The main circle has radius R and the circumscribing circle has radius $R + r$.

Because of this special configuration, next to my choice of origin and x-axis, the equations for the centre of the circle which touches the outer circle and the main circle, parameterized over its radius δ , simplify into

$$x = R + r - \delta \frac{2R + r}{r}$$

$$y^2 = (R + r - \delta)^2 - x^2.$$

Limit situations are $(x, y) = (R + r, 0), (-R, 0)$, for $\delta = 0, r$.

The requirement for touching the 3^{rd} circle, $\{x_i, y_i, r_i\}$, yields the equation in δ , with (x, y) given by the above formulas²⁸

$$(x - x_i)^2 + (y - y_i)^2 = (r_i + \delta)^2.$$

By changing the meaning of the 3^{rd} circle repeatedly – the just determined circle becomes the next – the set of shrinking touching circles, suggestively infinitely, has been obtained.

The above considerations and ideas have been coded in POSTSCRIPT as follows.

```

%!PS-Adobe- Circle with circles, cgl Mrt 97
%%BoundingBox: -150 -150 150 150
/R 100 def /r 50 def /Rr R r add def
/xi R neg def /yi 0 def /ri r def
Rr 0 moveto r 0 R 0 360 arc
Rr 0 moveto 0 0 Rr 0 360 arc
r neg 0 moveto xi yi r 0 360 arc
stroke
%tangent circle: x, y, d(delta)
/x{Rr d 2 R mul r div 1 add mul sub}def
/y{Rr d sub dup mul x dup mul sub sqrt}def
/fd{ri d add
x xi sub dup mul
y yi sub dup mul add sqrt sub}def
/solveit{%invariant: 1 fd > 0 and u fd <= 0
/d .5 1 u add mul def
fd 0 lt{/l d def}
{/u d def}ifelse
u 1 sub eps gt{solveit}
{/d .5 1 u add mul def}ifelse
}def
%

```

28. The concise and implicit equation for δ is suited for solving by computer. The explicit solution was communicated by H.J. van de Stadt and is due to Soddy. Another approach for obtaining an explicit formula for δ in terms of a square root and the arithmetic operations is by formula manipulation programs. I'm not sure whether these programs would yield Soddy's elegant representation.

```

/eps .01 def /lw 1 def %decrease linewidth
25{/l .25 ri mul def /u ri def solveit
  x d add y moveto x y d 0 360 arc
  x d add y neg moveto x y neg d 0 360 arc
  /lw lw .035 sub def lw setlinewidth stroke
  /ri d def /xi x def /yi y def
}repeat %showpage

```

Remarks.

I coded a –primitive, quick and dirty– zero-finding (bisection) operator in POSTSCRIPT.²⁹

Interesting is to extend tiling to the hyperbolic plane à la Coxeter, or to variate the circular arc by a pleasing line, for example anthropomorphically à la Escher.

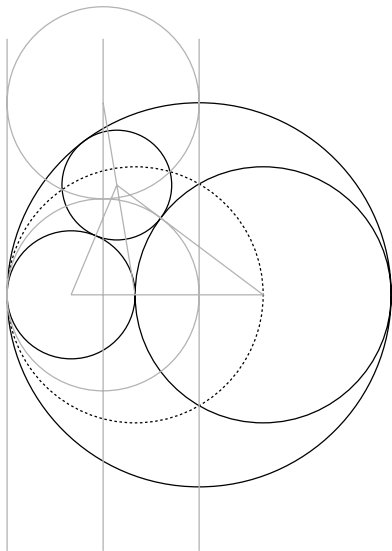
To write a program to fill up all (3-sided) cusps by circles, or all the new circles, infinitely, is interesting, and a nice, but not trivial, exercise in recursive programming.³⁰

An explicit solution for the radius of the 4th inscribed circle due to Soddy was communicated when the paper was in proof by H.J. van de Stadt and reads as follows.

$$\frac{1}{\delta} = \frac{1}{r_1} + \frac{1}{r_2} + \frac{1}{r_3} + 2\sqrt{\frac{1}{r_1 r_2} + \frac{1}{r_2 r_3} + \frac{1}{r_3 r_1}}$$

I decided to let the POSTSCRIPT code for this drawing unaltered. The use of the zero finding in POSTSCRIPT for solving an equation might be of use in other contexts.

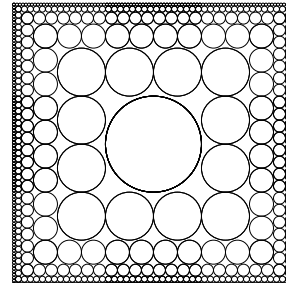
Pondering about this problem made me realize that times have changed. We don't need anymore for these kinds of problems solutions by compass and ruler, or solutions by transformation techniques such as the inversion in a circle.³¹ With our fancy computers approaches akin to this tool are emerging, and IMHO, with all respect, the above is such an approach. Apollonius revisited, aha.



Example (The old approach)

For the interested reader a picture which exhibits the inversion method is included on the left. The bold circles are what we are talking about. The thin circles and lines are the inverted ones, with the dashed circle the circle of inversion.

Example (Squares limits variant with circles)

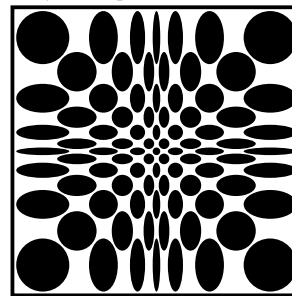


6.5 Metamorphoses

Escher played with changing the patterns while tiling. Schröfer did this with changing a circle and his result is captivating, with a nice Op Art effect. Vasarely³² has exploited this in depth, and used colors with breathtaking results.

Hofstadter³³ calls this parquet deformations.

Example (Schröfer's Op Art)



Note how the circle in the center grows when moving out-bound, and in general becomes an ellipsis. The vertical and

29. Maybe, I should recast in POSTSCRIPT in due time the famous zero-in algorithm developed at the CWI, where a combination of strategies – bisection, interpolation and extrapolation – have been implemented.

30. But, ... maybe there is another inroad when we look upon it as a fractal. H.J. van de Stadt mentions that this fractal is known as Pharaoh's Breastplate, and can be found in the book on fractals by Mandelbrot. In the note on fractals I'll come back on Soddy's formula and the fractal.

31. The approaches from the old days are discussed in for example Courant and Robbins: What is mathematics? OUP, New York, ISBN 0-19-502517-2.

32. A Hungarian computer artist.

33. Hofstadter, D R (): Metamagical themas – questioning for the essence of mind and pattern.

horizontal axes of the ellipses are proportional to the abscis and ordinate of its position.

```

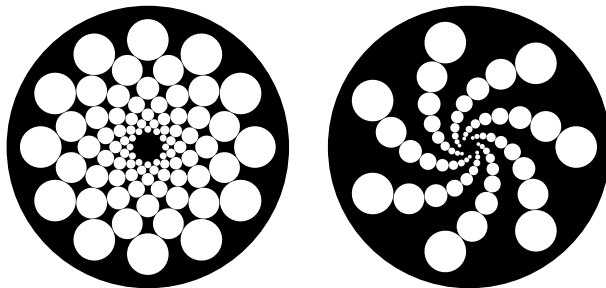
%!PS-Adobe Schroefers Op Art, cgl Nov 96
%%BoundingBox: -190 -190 190 190
/s 5 def
/drawgc{gsave r c translate
  r abs 5 div s add
  c abs 5 div s add scale
  0 1 moveto 0 0 1 0 360 arc fill
  grestore
}def
/schrofer{/flipflop true def
/indices[30 21 14 9 5 2 0
  -2 -5 -9 -14 -21 -30]def
indices{/r exch s mul def gsave
indices{/c exch s mul def
  flipflop{drawgc}if
  /flipflop flipflop not def
}forall grestore}forall
-38 s mul dup moveto 0 76 s mul rlineto
76 s mul 0 rlineto 0 -76 s mul rlineto
closepath 5 setlinewidth stroke
}def
schrofer %showpage

```

Remark. In the coding the use of an array for the indices, next to forall, and the flipflop mechanism are interesting.

My METAFONT code is similar, modulo some syntactic sugar.

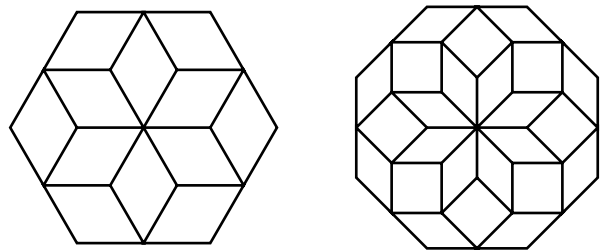
Example (*Variants*)



7 Tiling by diamonds

The following example allows a nice demonstration of how to use the ‘Turtle graphics’ idea for coding in POSTSCRIPT.

Example (*Division of hexagon and octagon*)



Just diamonds, a star or cubes? Isn’t it nice that hexagons can be divided so beautifully into 12 diamonds?

```

%!PS-Adobe- Diamonds, cgl Feb 97
%%BoundingBox: -50 -42 175 42
/turtle{%direction and stepsize on stack
  /step exch def /dir exch def
  currentpoint translate
  dir rotate step 0 lineto
}def
/r 25 def 2 setlinejoin
6{r 0 moveto -60 r turtle
  120 r turtle
  60 r turtle
  120 r turtle
  -60 r turtle
  -120 0 turtle
}repeat stroke
%
125 0 translate /r .75 r mul def
8{r 0 moveto 45 r turtle
  -90 r turtle
  135 r turtle
  45 r turtle
  135 r turtle
  -90 r turtle
  45 r turtle
  -180 0 turtle
}repeat stroke %showpage

```

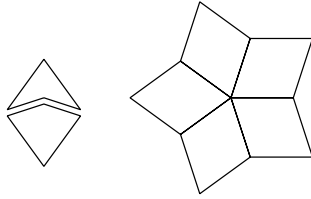
Explanation. turtle draws a line, parameterized over size and direction.

Note the use of a zero step at the end, next to the direction.³⁴

Example (*Penrose’s kites and darts*)

Related to dividing hexagons and octagons into diamonds is Penrose’s diamond, which he divided into 2 to cover the plain by its parts in a nonperiodic way.

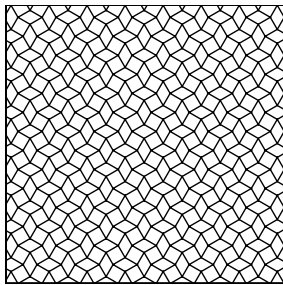
³⁴ Maybe I should have done this a little differently.



Remark. It's interesting to write a program which generates automatically tilings by these kites and darts. That is to teach a computer how to puzzle.

Example (*Rhombus as basic element*)

I chose a diamond as basic element. Of course one can also code the wiggly lines.



The above is obtained as follows

```

%!PS Tiling fourthree, cgl 96
%%BoundingBox: -100 -100 100 100
/a 10 def /ha a .5 mul def
/tile {% rhombus + 90 rotated rhombus
  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
/tena a 10 mul def
/frame {tena neg tena moveto
  tena 2 mul 0 rlineto 0 tena -2 mul rlineto
  tena -2 mul 0 rlineto closepath}def
/dotiling{a -11 mul tena neg translate
  9{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

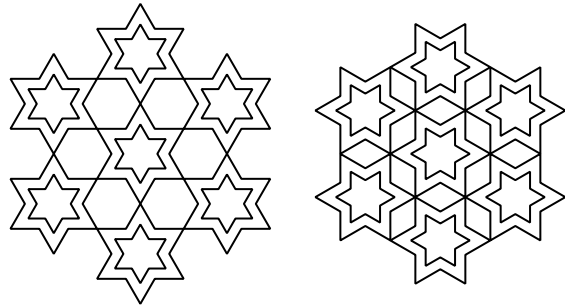
```

```
frame clip dotiling %showpage
```

My METAFONT code reads similar.

8 Tiling by stars

The following –non-tight ‘hexagon’ tilings –reminds me of tilings by the Moors.



```

%!PS-Adobe 6-stars, cgl Feb 97
%%BoundingBox: -70 -70 70 70
/r 25 def 2 setlinejoin
/six{%r on stack
  /rloc exch def
  rloc 0 moveto
  6{30 rotate .577 rloc mul 0 lineto
    30 rotate rloc 0 lineto}repeat
  closepath stroke
}def
30 rotate
r six .588 r mul six
6{gsave 2 r mul 0 translate
  r six .5 setlinewidth .588 r mul six
  grestore
  60 rotate}repeat %showpage

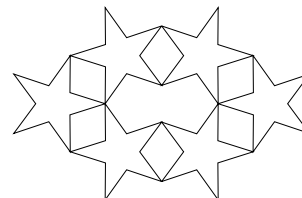
```

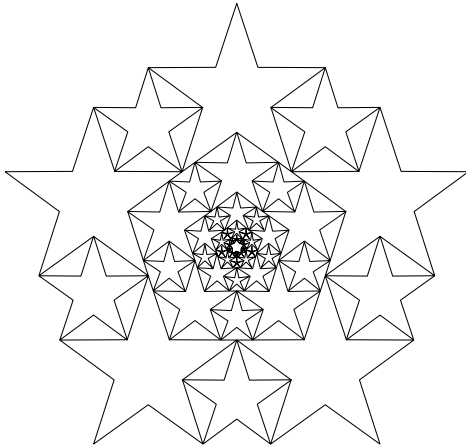
Looking at the left picture I found it hard to recognize that it was just a tiling of 6-stars. In reality the tessallations also exhibit ‘hidden’ lines.

Example (*Dual of composition of pentagons I*)

Stars and the enveloping polygons are related.

As exercise for the reader the following dual of the composition of pentagons I.



Example (*Composite of pentagrams*)

This code arose after Jackowski's visit to NTG's fall 1996 meeting, where he lectured and dealt with stars among other things.

```

%!PS-Adobe- Tiling pentagrams, cgl Nov 96
%%BoundingBox: -100 -100 100 100
/star{%n r on stack, n>=5
  /rstar exch def /nstar exch def
  /alfahstar 180 nstar div def
  /rinstar rstar 2 div alfahstar cos
    mul 1.5 sub def
  0 rstar moveto
  nstar{alfahstar rotate 0 rinstar lineto
    alfahstar rotate 0 rstar lineto
  }repeat closepath stroke
}def
%
/n 5 def /r 5 def          %(r,n)-gon
/alfa 360 n div def
/lw .1 def 2 setlinejoin
%loop over the 'rings'
4{lw setlinewidth
  /rh r 2 div          def %r half
  /rin r 36 cos mul def %r inside
  %big stars
  /c rin 36 cos mul 2 mul def
  n{gsave 0 c translate n rin star
    grestore alfa rotate
  }repeat
  %'half-sized' stars
  gsave 36 rotate
  /c rin rh add def
  n{gsave 0 c translate 36 rotate
    n rh star
    grestore alfa rotate

```

```

  }repeat
  grestore
  /r 36 cos dup 2 mul 1 add mul r mul def
  /lw lw .2 add def
}repeat %showpage

```

Explanation. The figure is composed of rings of stars, with in each ring stars of 2 sizes. The bigger stars circumscribe a spurious pentagon. The radius r of (the circumscribing circle of) this pentagon is taken as the independent parameter of the drawing. Dependent quantities are the following.

Radius inner circle: $r_{inside} = r \cos 36$

Side pentagon: $2r \sin 36$

Bigger star

radius: $r \cos 36$

distance from center: $2r \cos^2 36$

Smaller star

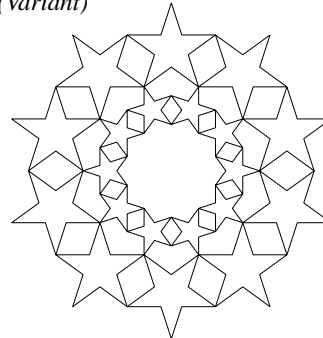
radius: $.5r$

distance from center: $r_{inside} + .5r$

The radius of the next ring: $r \cos 36 (2 \cos 36 + 1)$

BoundingBox:³⁵ $r (\cos 36 (2 \cos 36 + 1))^n \approx 2.12^n r$.

Drawing a star has been explained in 'Stars around I.'³⁶ Note the use of 2 setlinejoin to circumvent spurious sharp corners.³⁷

Example (*Variant*)**9 Acknowledgements**

Thank you Bogusław Jackowski for your suggestions, inspiring examples, and help. Thank you Sasha Berdnikov for providing me with examples of tilings. Thank you Erik Frambach for comments and discussions. Peter van Summeren pointed me to some literature items. Denis Roegel informed me about Truchet's tiling and the GUTenberg

35. Radius of circumscribing pentagon, with n the number of rings.

36. See MAPS 97.1. The idea is that the star can be seen as a pentagon with a broken line as side. The coordinates of the junction of the 2 line elements of the broken line follow from a backside of the envelope calculation.

37. Courtesy Marcel Tünnissen, who also played with the figure and applied technology from 3D—extending the sides of polyhedra to yield starred figures—to 2D.

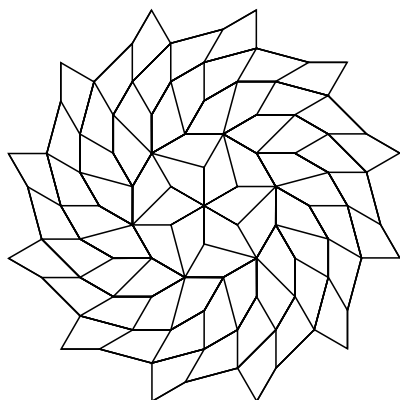
contest on the issue, next to providing me with a copy of a GUTenberg article about the winners of the contest. H.J. van der Stadt communicated Soddy's formula.

Many examples have been borrowed from the earlier mentioned book by Grünbaum and Shephard.

As usual Jos Winnink proofed the paper and lent a helping hand in procrusting the article into MAPS outfit. Thank you all.

10 What more?

The tilings, as overwhelming they might seem, are just the top of the iceberg. What about tiling on curved surfaces, for example on a sphere, as well as general tiling in 3D? Maybe, I will come back on the issue when computers have real holographic 3D viewing possibilities.



11 Conclusions

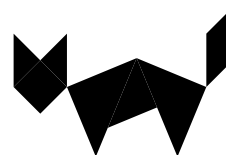
Just the use of a few graphical primitives in POSTSCRIPT and awareness of the user versus device space can yield interesting pictures via little POSTSCRIPT code. Sometimes it is handier and more elegant to use METAFONT as a declarative language. On the other hand METAFONT's pictures can't be rotated over arbitrary angles.

Differences in the concept of the path data structure in POSTSCRIPT and METAFONT entails different programs. The most apparent differences are in tiling and clipping. I would welcome in POSTSCRIPT an operator similar to METAFONT's point (*expression*) of (*path*).

Happily the same splines are used underneath facilitating METAFONT to calculate the splines from a declarative specification ready for use in POSTSCRIPT.

I never realized that tiling, especially when done by computer, can be so amazing. So far I've no access to color printers, alas. Undoubtedly a mer a boire awaits me there.

I hope that the given codes will contribute to the literature of METAFONT and POSTSCRIPT codes. I welcome comments.



12 Appendix: Postponed codes

12.1 Mondrian's Two lines

In the original Mondrian the two lines did not divide the sides by the golden ratio. An oversight by Mondrian?

In principle a very simple code, but becomes interesting with proper coloring of the background and with lines of non-negligible thickness.

```

%!PS-Adobe- Mondrian's Two lines, cgl 96
%%BoundingBox: -50 0 50 100
/D 50 def
/R D 1.4142 mul def %side
/gr .618 R mul def
/delta 10 def /mdelta delta neg def
45 rotate
0 0 moveto R 0 lineto
R R lineto 0 R lineto
closepath
gsave .985 setgray fill grestore
clip newpath
%gr 0 moveto 0 gr lineto
gr delta add mdelta moveto
mdelta gr delta add lineto
%gr R moveto 0 R gr sub lineto
gr delta add R delta add moveto
mdelta R gr sub delta sub lineto
R 30 div setlinewidth stroke %showpage

```

Explanation. I chose for a clipping boundary and to let the thick lines extend to be cut appropriately.

12.2 Yin-Yang

An exercise in using (circular) arcs.

METAFONT code.

```

%Yin-Yang, cgl Jan 97
size=100;
fill ((halfcircle&
      halfcircle rotated 180 scaled.5
      shifted(-.25,0)&
      reverse (halfcircle scaled.5
      shifted(.25,0))&
      cycle)scaled size);
draw fullcircle scaled size;
fill fullcircle scaled .125size
      shifted (.25size,-.0625size);

```

```
unfill fullcircle scaled .125size
      shifted (-.25size, .0625size);
showit; end
```

Explanation. It is all about drawing, *casu quo* (un)filling, of (arcs of) circles, and to scale and reposition these appropriately. In plain METAFONT we have `half/fullcircle` which have to be scaled, rotated, and positioned. In POSTSCRIPT a circle macro is easily written and used similarly. Interesting is the use of `reverse` in METAFONT in order to create closed contours, which in POSTSCRIPT is done implicitly via `arcn`, that is the arc in negative direction.

A similar POSTSCRIPT code reads as follows.

```
%!PS-Adobe- Yin-Yang, cgl 96
%%BoundingBox: -50 -50 50 50
/R 50 def      /mR R neg def
/r R 8 div def /mr r neg def
/hR R 2 div def /mhR hR neg def
/circle{%center on stack
  translate
  r 0 moveto 0 0 r 0 360 arc
}def
R 0 moveto 0 0 R 0 180 arc
      mhR 0 hR 180 360 arc
      hR 0 hR 180 0 arcn
fill
gsave mhR r circle
      1 setgray fill grestore
gsave hR mr circle
      0 setgray fill grestore
R 0 moveto 0 0 R 0 360 arc stroke %showpage
```

The following POSTSCRIPT code was output by MetaPost.³⁸ Some ≈ 50 lines of `curveto`, `fill` and `stroke`. Much less readable than my straight (handcoded) PostScript, moreover, ‘structure’ has disappeared.

```
%!PS-Adobe- Yin-Yang, MetaPost (JJW)
%%BoundingBox: -51 -51 51 51
newpath 50 0 moveto
50 13.26 44.73 25.98 35.35 35.35 curveto
25.98 44.73 13.26 50 0 50 curveto
-13.26 50 -25.98 44.73 -35.35 35.35 curveto
-44.73 25.98 -50 13.26 -50 0 curveto
-50 -6.63 -47.37 -12.99 -42.68 -17.68 curveto
-37.99 -22.37 -31.63 -25 -25 -25 curveto
-18.37 -25 -12.01 -22.37 -7.32 -17.68 curveto
-2.63 -12.99 0 -6.63 0 0 curveto
0 6.63 2.63 12.99 7.32 17.68 curveto
12.01 22.37 18.37 25 25 25 curveto
31.63 25 37.99 22.37 42.68 17.68 curveto
47.37 12.99 50 6.6350 0 curveto
closepath fill
```

```
0 2 dtransform truncate idtransform
      setlinewidth pop [] 0 setdash
1 setlinejoin 10 setmiterlimit
newpath 50 0 moveto
50 13.26 44.73 25.98 35.35 35.35 curveto
25.98 44.73 13.26 50 0 50 curveto
-13.26 50 -25.98 44.73 -35.35 35.35 curveto
-44.73 25.98 -50 13.26 -50 0 curveto
-50 -13.26 -44.73 -25.98 -35.35 -35.35 curveto
-25.98 -44.73 -13.26 -50 0 -50 curveto
13.26 -50 25.98 -44.73 35.35 -35.35 curveto
44.73 -25.98 50 -13.26 50 0 curveto
closepath stroke
newpath 31.25 -6.25 moveto
31.25 -4.59 30.59 -3.00 29.42 -1.83 curveto
28.25 -0.66 26.66 0 25 0 curveto
23.34 0 21.75 -0.66 20.58 -1.83 curveto
19.41 -3.00 18.75 -4.59 18.75 -6.25 curveto
18.75 -7.91 19.41 -9.50 20.58 -10.67 curveto
21.75 -11.84 23.34 -12.5 25 -12.5 curveto
26.66 -12.5 28.25 -11.84 29.42 -10.67 curveto
30.59 -9.50 31.25 -7.91 31.25 -6.25 curveto
closepath fill 1 setgray
newpath -18.75 6.25 moveto
-18.75 7.91 -19.41 9.50 -20.58 10.67 curveto
-21.75 11.84 -23.34 12.5 -25 12.5 curveto
-26.66 12.5 -28.25 11.84 -29.42 10.67 curveto
-30.59 9.50 -31.25 7.91 -31.25 6.25 curveto
-31.25 4.59 -30.59 3.00 -29.42 1.83 curveto
-28.25 0.66 -26.66 0 -25 0 curveto
-23.34 0 -21.75 0.66 -20.58 1.83 curveto
-19.41 3.00 -18.75 4.59 -18.75 6.25 curveto
closepath fill showpage
```

Remark. Note the redundancies, for example `newpath`.

12.3 Escher's sun and moon

I ‘scanned’ the picture and created appropriate paths in METAFONT by incorporating the points, specifying directions and using ordinary and splice joins. When finished METAFONT was ordered to write the curves—data for the spline pieces suitable for POSTSCRIPT—to the log file. This file was edited into a straight PostScript program. My poor man’s MFTOEPS, which is conceptually simple and does not require knowledge of clever tools.

The essential issues of the program are listed below.

```
path p[]; s=30;
p11=origin..(-.7s,.5s)..(-2s,0)..
      (-2.6s,.3s)..(-3.5s,0);
p12=point 4 of p11..
```

³⁸. Courtesy Jos Winnink, who adapted my METAFONT program for the purpose.

```

    {(1,3)}(-3.2s,s){(3,1)}..(-2.2s,s)&
    (-2.2s,1s){(-1,1)}..(-2.9s,1.9s)&
    (-2.9s,1.9s){(1,1)}..{right}(-1.5s,2.7s);
p13=point 6 of p12{(-1,-1)}..(-2s,2s)&
    (-2s,2s){(5,1)}..(-.6s,2s)&
    (-.6s,2s)..(-1.4s,1.4s)---
    (-1.6s,.9s)..(-s,.8s)..(.2s,1.2s)&
    (.2s,1.2s)..point 0 of p11;
p1= reverse(p11..p12..p13..cycle);
fill p1;
%write to log file suitable for PS
"Bird1";
for k=0 upto 24:
    show postcontrol k of p1;
    show precontrol k+1 of p1;
    show point k+1 of p1; "curveto";
endfor
%similar for other two (dark) birds
"spurious bird";
p4=point 0 of p33..(-2.2s,-2s)&
    (-2.2s,-2s){left}..(-3s,-2.1s)&
    (-3s,-2.1s)..(-2.5s,-.7s)&
    (-2.5s,-.7s)..point 4 of p11;
draw p4;
%write to log file suitable for PS
%similar for Bird 5 and 6
showit; end

```

Note. "*string*" writes the *string* to the log file.

The PostScript program is a concatenation of curveto-s.

```

%!PS-Adobe Escher's zon en maan; cgl Jan 97
%%BoundingBox: -100 -100 100 100
%Bird1
0 0 moveto
2 12 4 24 6 36 curveto
-4.34 28.13 -17.00 23.91 -30 24 curveto
%etc.

```

I guess the Sun and Moon picture could have been drawn easily by WYSIWYG graphics X-works software with a request for POSTSCRIPT output?

12.4 The squares patterns

Without explanation the codes for the simple squares patterns exercises are included. I hope it is clear enough. For the accurate calculation of the BB a little knowledge of geometry is needed.

```

%!PS-Adobe- Kepler's squares, cgl Dec 96
%%BoundingBox: -50 -37.5 50 50

```

```

/r 25 def %r dodecahedron
/s r 2.8284 75 sin mul div def
/d r 2 mul s .707 mul sub def
/rotsq{gsave s 0 moveto
    4{0 s lineto 90 rotate}repeat
    closepath stroke
    grestore
}def
/dodeca{6{gsave r s sub 0 translate
    rotsq
    grestore 60 rotate
    }repeat
}def
/rstar s s mul r s sub dup mul add sqrt def
3{gsave 0 rstar translate dodeca grestore
    120 rotate
    }repeat %showpage

```

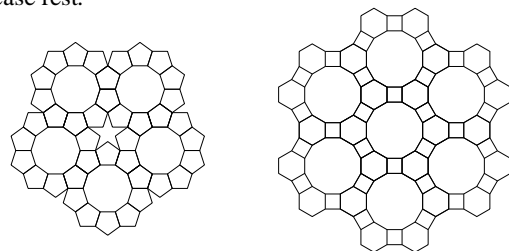
Interesting is the use of scaling in the code below.

```

%!PS-Adobe- More Square madness, cgl Feb 97
%%BoundingBox: -80 -80 80 80
/r 50 def
/s .5858 r mul def /ms s neg def
/sds .707 s mul def /msds sds neg def
/element{gsave r 0 translate
    sds 0 moveto 0 sds lineto
    msds 0 lineto 0 msds lineto
    closepath fill
    grestore
}def
/f r sds sub r sds add div def
/gl 1 def
%
3{8{element 45 rotate}repeat
    f f scale /gl f gl mul def
    gl setgray
    }repeat %showpage

```

My case rest.



By the way, the right figure is classified by {4, 6, 12}, a nice layout for a herb garden. Have fun, and all the best.

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

abstract

Er lopen in de wereld zo veel \TeX -gebruikers rond, dat het zelden zal voorkomen dat een normale gebruiker de eerste is die een probleem tegen komt. Meestal zal dit probleem ook al door iemand anders zijn opgelost en vaak staat de oplossing op de 4all \TeX , \TeX live of een andere CD-ROM. Indien daar niet te vinden is staat er vast wel een oplossing op CTAN¹ of één of andere Internetsite. Het probleem zit hem niet in de beschikbaarheid, maar in het opsporen van de oplossing. Artikelen in MAPS zijn daarvoor natuurlijk een belangrijke informatiebron. Sommige oplossingen zijn echter zo eenvoudig dat er nooit een artikel aan gewijd zal worden. Over een draadstriptang valt nu eenmaal niet al te veel te vertellen. In Toolbox zal niet alleen de draadstriptang worden besproken, maar ook het gebruik van lucifers om het plastic van het einde van het snoer af te branden. Stuur ook informatie over uw favoriete tools naar mij op, zo kan Toolbox een belangrijke informatiebron worden over utilities, macro's en andere eenvoudige trucs voor de gewone \TeX -gebruiker. Deze keer is de spil van het verhaal PostScript.

Van \TeX naar \TeX

Het gebeurt maar al te vaak dat je in het ene \TeX -document een ander \TeX -document wilt opnemen. Vaak kan dit door direct de benodigde informatie uit het ene bestand naar het andere te kopiëren. Wanneer er echter gebruik wordt gemaakt van aparte stijlbestanden of plain \TeX in \LaTeX moet worden opgenomen (en omgekeerd) gaat dit niet zo gemakkelijk. Het klassieke voorbeeld is de opname van een vragenlijst als een bijlage in een proefschrift. Het is natuurlijk onze eer te na om in zo'n geval met schaar en plakband of een scanner te gaan werken. Gelukkig is dit ook niet nodig, \dvips zelf biedt de oplossing. Met behulp van de `-E` optie maakt \dvips een `encapsulated postscriptfile` aan die daarna gewoon als afbeelding in het andere document kan worden opgenomen.² Helaas werken de opties `-S` en `-i` niet goed samen met de `-E` optie. In MS-DOS zullen de afbeeldingen dus als volgt moeten worden aangemaakt (we gaan even uit van een document `doc.dvi` van 4 pagina's):

```
for %a in (1 2 3 4) do dvips32 -E -p %a -l %a \  
-o doc%a.eps
```

Na het uitvoeren van dit commando zijn de bestanden `doc1.eps`, `doc2.eps` etc. aangemaakt, met `\includegraphics{bestandsnaam}` kunnen deze vervolgens in het document worden opgenomen.

GhostScript

De oplossing in de vorige paragraaf maakt wel gebruik van PostScript—een printerbesturingstaal. Nu zal niet iedereen over een PostScript-printer beschikken. Gelukkig bestaat het programma GhostScript: een gratis programma dat op vrijwel ieder operating system beschikbaar is. GhostScript kan PostScript-bestanden afbeelden en is in staat deze bestanden op de meeste gangbare printers af te drukken. GhostScript is te vinden op `http://www.cs.wisc.edu/~ghost/`. Als u met Windows werkt zult u waarschijnlijk ook GhostView, een Windows-interface voor GhostScript, willen downloaden.

Een belangrijk nadeel van GhostScript is dat het geen gebruik kan maken van de lettertypes die in de printer zitten. Indien er alleen met Computer Modern wordt gewerkt is dit natuurlijk geen probleem, maar bij voorbeeld Times en Helvetica zullen niet helemaal juist worden weergegeven. Omdat op deze lettertypes copyright zit worden ze niet met GhostScript meegeleverd. Gelukkig levert Adobe (de leverancier van ondermeer PostScript en deze lettertypes) hier de oplossing: met Acrobat Reader, een ander product van Adobe, worden Times en Helvetica meegeleverd. We moeten alleen nog aan GhostScript vertellen waar de letters gevonden kunnen worden. Dit gebeurt door de regels in het bestand `Fontmap` die verwijzen naar Times en Helvetica conform figuur 1 aan te passen. Eventueel kan voor de bestandnamen nog een path worden opgenomen, bijvoorbeeld `(c:\acrobat\fonts\TIR____.pfb)`, voor het lettertype Times Roman

GhostScript is zo'n uitgebreid programma, dat we er in komende afleveringen van deze column nog herhaaldelijk op terug zullen komen.

1. CTAN is het Comprehensive \TeX Archive Network, de dichtsbijzijnde CTAN-site is `ftp://ftp.cs.ruu.nl/pub/tex-archive`.

2. In verband met het mogelijk vergroten/verkleinen van deze afbeelding verdient het wel aanbeveling om Type 1 fonts te gebruiken, maar deze zijn tegenwoordig gelukkig ook voor de Computer Modern fonts zonder problemen beschikbaar. Ze zijn te downloaden uit de directory `fonts/cm/ps-type1/bluesky` van CTAN.

```

%/Courier          /NimbusMonL-Regu      ;
%/Courier-Oblique /NimbusMonL-ReguObli    ;
%/Courier-Bold    /NimbusMonL-Bold      ;
%/Courier-BoldOblique /NimbusMonL-BoldObli    ;
/Courier          (COM____.pfb);
/Courier-Oblique  (COO____.pfb);
/Courier-Bold     (COB____.pfb);
/Courier-BoldOblique (COBO____.pfb);

%/Helvetica       /NimbusSanL-Regu      ;
%/Helvetica-Oblique /NimbusSanL-ReguItal  ;
%/Helvetica-Bold   /NimbusSanL-Bold      ;
%/Helvetica-BoldOblique /NimbusSanL-BoldItal ;
/Helvetica        (HV____.pfb);
/Helvetica-Oblique (HVO____.pfb);
/Helvetica-Bold   (HVB____.pfb);
/Helvetica-BoldOblique (HVBO____.pfb);

%/Times-Roman     /NimbusRomNo9L-Regu    ;
%/Times-Italic    /NimbusRomNo9L-ReguItal ;
%/Times-Bold      /NimbusRomNo9L-Medi    ;
%/Times-BoldItalic /NimbusRomNo9L-MediItal ;
/Times-Roman      (TIR____.pfb);
/Times-Italic     (TII____.pfb);
/Times-Bold       (TIB____.pfb);
/Times-BoldItalic (TIBI____.pfb);

```

Figuur 1. Aanpassingen aan het bestand Fontmap van GhostScript. De oorspronkelijke tekst is telkens met %-tekens uitgezet.

PSutils

Als we het over PostScript hebben, moeten we ook de PSutils noemen, een serie hulpprogramma's om allerlei bewerkingen op PostScript-bestanden uit te voeren.³ Ik gebruik zelf vrijwel dagelijks psnup. Met dit programma is het mogelijk meerdere pagina's uit een PostScript-bestand op één pagina af te drukken. Het commando `psnup -pa4 -nup 2 bestand1.ps bestand2.ps` is equivalent aan naar de kopieermachine lopen en twee A4-tjes verkleind naar één A4-tje kopiëren.

Een tweede handig programma uit de PSutils is psbook, dat gebruikt kan worden om de pagina's uit een PostScript-bestand zo te rangschikken dat ze afgedrukt kunnen worden als een boekje. Handig voor verenigingsblaadjes en ander folder-achtig materiaal. Rond het aantal pagina's *naar boven* af op een veelvoud van vier (er komen immers op ieder dubbelzijdig bedrukt A4-tje vier pagina's te staan). Stel dat we 18 pagina's hebben, dan zal het boekje uiteindelijk 20 pagina's tellen: type dan `psbook -s20 bestand1.ps bestand2.ps` en de pagina's

worden op de juiste manier gerangschikt.

Daarnaast zijn er `pselect` dat gebruikt kan worden om willekeurige pagina's uit een PostScript-bestand te selecteren en `pstops`, waarmee zo ongeveer iedere willekeurige rangschikking van de pagina's uit een PostScript-bestand mogelijk is.

And now something completely different

Genoeg over PostScript, nu eens iets anders. 'Nu eens iets anders,' hoe vaak denk je dat niet als beginnende (of zelfs gevorderde) L^AT_EX-gebruiker. Met `\usepackage{times}` het lettertype veranderen wil nog wel lukken, maar het zelf maken van verder gaande lay-out veranderingen vereist meteen wel erg veel expertise. Gelukkig is het ook mogelijk te profiteren van het werk van anderen: gewoon een kwestie van het traditionele `\documentclass{article}` vervangen door een ander type document. Mijn favoriet zijn de reference manual classes van Partl en Kielhorn.⁴ De reference manual macro's bestaan uit twee nieuwe documentclasses: `refart` en `refrep` ter vervanging van respectievelijk de `article` en de `report` class. Een voorbeeld van een met deze stijl gemaakt document staat in figuur 2. Alhoewel de reference classes, de naam zegt het al, bedoeld zijn voor handleidingen, zijn ze wat mij betreft ideaal voor syllabi en proefversies van artikelen. In de preamble voeg ik alleen altijd even de commando's `\setlength{\parindent}{\parskip}` en `\addtolength{\parindent}{\baselineskip}` toe. Ik hou er niet van als de eerste regel van mijn alinea niet inspringt.⁵

Het installeren van packages

Dit is misschien het juiste moment om iets te zeggen over het installeren van nieuwe packages (zoals het reference manual package). Packages bestaan over het algemeen uit één of meer `.dtx` en één of meer `.ins` files (in het geval van `refman` zijn er twee bestanden: `refman.dtx` en `refman.ins`). Daarnaast zal er over het algemeen een bestand met instructies te vinden zijn waarvan de naam lijkt op `readme`, `install.txt` of de extensie `.upl` heeft. Verder is er soms nog wat 'troep'; in het instructiebestand zal meestal wel aangegeven worden waar deze troep voor dient. Plaats al deze bestanden in een lege, tijdelijke directory en lees het instructiebestand (indien aanwezig).

Voor het vervolg van de installatie moeten we weten

3. De PSutils zijn op CTAN te vinden in de directory `/support/psutils`.
4. Deze macros zijn te downloaden uit de directory `macros/latex/contrib/supported/refman` van CTAN.
5. De werkwijze is niet geheel netjes, eigenlijk zou ook de maxipage-omgeving aangepast moeten worden.

4 Het uitvoeren van het onderzoek	26
4.1 Literatuur verzamelen	26
4.1.1 Social Science Citation Index	27
4.1.2 CD-Rom en andere elektronische informatie	27
4.1.3 Bronnen van literatuurverwijzingen	27
4.2 De bibliotheek-catalogus	28
4.3 Het verzamelen van gegevens	29
4.3.1 Het kiezen van een onderzoeksmethode	29
4.3.2 Het gebruik van openbare gegevens	30
4.3.3 Het houden van interviews	30
4.3.4 Overige methoden	31
4.4 Analyseren van gegevens	31
A Relevante tijdschriften	33
A.1 Financial en Management Accounting	33
A.2 Bestuurlijke informatickunde	36
A.3 Financiering	37
A.4 Marketing	37
A.5 Organisatie en leiding	38
B Vakgroepsspecifieke informatie	39
B.1 Bestuurlijke Informatickunde	39
B.2 Vakgroep Financiering	39
B.2.1 Scriptie schrijven	39
B.2.2 Stagebegeleiding	41
B.3 Vakgroep Kosten- en Winstvraagstukken	41

1 Inleidende opmerkingen

Het doel van deze handleiding is het helpen van studenten economie bij het doen van het onderzoek voor hun afstudeerscriptie en het schrijven van deze scriptie. Geprobeerd wordt om voor het gehele scriptietraject aanbevelingen te doen. Deze aanbevelingen zijn over het algemeen geen wetten. Zo wordt er in hoofdstuk 3 een systeem gesuggereerd om literatuurverwijzingen in de scriptie op te nemen. Dit systeem is niet zaligmakend. In een aantal omstandigheden zal het praktischer zijn om een ander systeem te gebruiken. Dit is dan ook zonder meer toegestaan, mits het gehanteerde systeem consequent wordt toegepast en verdedigbaar is. Gezond verstand heeft voorrang boven de in de handleiding gegeven richtlijnen. Wel dient de lezer zich te realiseren dat sommige scriptiebegeleiders hun eigen eisen stellen aan schrijfstijl, lay-out en wijze van citeren. Het is aan te raden hiernaar in een vroeg stadium van het schrijven van de scriptie te informeren. Wie op zoek is naar gedetailleerdere richtlijnen kan bij voorbeeld de *Publication Manual of the American Psychological Association* of de *Chicago Manual of Style* raadplegen.

dit is een voorbeeld

Af en toe worden in deze handleiding voorbeelden gebruikt ter verheldering van het besprokene. Kleine voorbeelden worden in de marge geplaatst, zoals hier (links) is getoond. Grotere voorbeelden worden tussen twee lijnen over de volle breedte van de pagina geplaatst zoals hieronder wordt getoond:

Dit is een voorbeeld dat te groot is om in de marge geplaatst te worden. Derhalve wordt het tussen twee lijnen over de volledige breedte van de pagina geplaatst.

1.1 Tijdsbeslag

Officieel zijn in het studieprogramma voor het schrijven van een scriptie 13 studiepunten gereserveerd. Dit houdt echter niet in dat je bij het plannen van je studie ervan uit kunt gaan dat je in 13 weken tijd je scriptie kunt afronden. Een eerste reden waarom dit niet mogelijk is, is gelegen in de begeleiding. Tijdens het scriptietraject zal je scriptiebegeleider je scriptie-opzet dienen te beoordelen. Dit kan als snel een week kosten. Wanneer de scriptie af is, zal de begeleider deze ook nog dienen te lezen teneinde een eindoordeel te vormen. Ook dit kost enige tijd. Ten slotte organiseren sommige vakgroepen seminars, waarin je je scriptie dient te presenteren, alvorens een cijfer wordt toegekend. Voordat je af kunt studeren, dient er zo'n bijeenkomst plaats te hebben gevonden. Het kan voorkomen dat je een paar weken moet wachten op de volgende bijeenkomst.

Figuur 2. Voorbeeld van pagina-opmaak met behulp van de reference classes

waar $\text{T}_{\text{E}}\text{X}$ naar invoerbestanden zoekt. Als het goed is staat dit in de local guide. Compileer anders een eenvoudig document en kijk in het `.log`-bestand. Hier valt te lezen uit welke directories $\text{T}_{\text{E}}\text{X}$ bestanden leest. Staat hier bij voorbeeld dat het bestand `article.cls` wordt ingelezen uit de directory `c:\emtex\texinput\base`, dan worden hoogstwaarschijnlijk alle subdirectories van `c:\emtex\texinput` door $\text{T}_{\text{E}}\text{X}$ afgezocht. Maak in de gevonden directory een nieuwe directory aan. In ons geval wordt dit `c:\emtex\texinput\refman`. Dit wordt de directory waarin we de bij het package horende bestanden uiteindelijk gaan plaatsen.

Deze bestanden moeten we echter eerst aanmaken. Tenzij in het instructiebestand anders staat aangegevend doen we dit door $\text{T}_{\text{E}}\text{X}$ te runnen met het `.ins` bestand als invoer. Voor de `refman`-bestanden geven we dus het commando `tex refman.ins`. De bestanden worden nu aangemaakt. Nadat de bestanden dit gebeurt is, plaatsen we alle bestanden met de extensies `.cls`, `.clo`, `.sty`, `.fd`, `.def` en `.cfg`

(voor zover aanwezig) in de zojuist aangemaakte directory. Het package is nu geïnstalleerd, we moeten alleen nog de documentatie aanmaken. Dit doen we door $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ te runnen op het `.dtx`-bestand. In ons geval: `latex refman.dtx`.⁶ Na het uitprinten kunnen de documentatie en de oorspronkelijke bestanden in aparte documentatie- en distributiedirectories worden geplaatst, of gewoon worden gewist.

Volgende keer

In het volgende nummer van MAPSwordt in Toolbox met name aandacht besteed aan literatuurverwijzingen.

6. Wie dit met $\text{emT}_{\text{E}}\text{X}$ doet krijgt de mededeling `TeX capacity exceeded`. Gebruik de optie `-ms 1200`. Het totale commando wordt dan in mijn geval: `tex386 -mt23800 -ms1200 &latex refman.dtx`.

Bijlage 7

TUG'97 conferentie

Erik Frambach

abstract

Een kort verslag van de conferentie van de T_EX Users Group 1997 in San Francisco, USA.

keywords

conferentie, TUG, San Francisco

T_EX Comes Home

De jaarlijkse conferentie van TUG vond dit jaar plaats in San Francisco. Dat is dus erg dicht bij de oorsprong van T_EX, Stanford University. Het thema voor deze conferentie was daarom "T_EX Comes Home".

TUG had tot voor kort een eigen *office* in San Francisco. Die office werd echter net op dat moment opgeheven. Een beetje wrang, maar wel een verstandige beslissing. Voor de deelnemers aan de conferentie leverde dat een aardige verrassing op. Alle boeken, T-shirts, mokken, oude TUGboats en andere spullen die bij het ontruimen van de office overbleven werden voor een habbekrats te koop aangeboden.

De locatie

De conferentie vond plaats in de Lone Mountain Conference Center op de campus van de University of San Francisco. In de conference center had TUG een fraaie zaal om koffie en thee te drinken tijdens de pauzes, en om de lunch te gebruiken. Ontbijt konden we krijgen in een soort mensa een paar honderd meter verderop. Na een of twee dagen hadden velen echter door dat je dat ontbijt net zo goed kon overslaan en wachten op de eerste koffiepauze. Daarbij werd namelijk ook een bijna compleet ontbijt geserveerd.

De zaal waar de lezingen werden gegeven was in verhouding nogal pover. Om alle circa 80 deelnemers een zitplaats te geven was de zaal domweg te klein. Ook de luchtverversing was daar duidelijk niet op berekend. Een geluidsinstallatie ontbrak geheel, wat voor mensen achterin de zaal soms lastig was.

Het komt echter niet veel voor dat alle deelnemers bij een lezing aanwezig zijn, dus er viel wel mee te leven.

Preprint van proceedings

De preprint van de proceedings vond ik eerlijk gezegd weggegooid geld. Die bestond namelijk uit enkele verre van complete of onafgewerkte stukken, plus een complete

handleiding van (dus niet artikel over) L^AT_EX2html. Het is te hopen dat de definitieve versie beter in elkaar zit.

De lezingen

Hierna volgt een kort overzicht van de lezingen die gegeven zijn op de bijeenkomst. Ik ben niet overal bij geweest, dus het is mogelijk dat hier of daar iets niet (helemaal) klopt. Excuses daarvoor.

Op de eerste dag vertelde John Plaice over 'Omega', de 16-bits versie van T_EX, die nog heel wat meer kan dan we van T_EX gewend zijn. Verder vertelde Ross Moore over 'L^AT_EX2html' en Anita Hoover over perikelen bij de omschakeling naar het gebruik van L^AT_EX 2_ε.

Op de tweede dag presenteerde Kristoffer Rose 'Xy-pic' als middel om 2-dimensionele tekeningen te maken die geanimeerd kunnen worden. Sebastian Rathz gaf een lezing over het maken van Metapost plaatjes. Eitan Gurari liet zien hoe je met 'DraT_EX' tekeningen kunt maken. Vervolgens liet Ross Moore nog een geavanceerde toepassing zien van 'Xy-pic'. Sebastian Tannert presenteerde 'CIRC', een pakket waarmee met eenvoudige commando's ingewikkelde stroomschema's kunnen worden getekend.

In de middag kregen we van Peter Breitenlohner te horen wat de stand van zaken is met betrekking tot 'ε-T_EX'.

Vaak waren de toekomst van T_EX (ε-T_EX, SGML, etc.) aanleiding tot discussies in de wandelgangen en soms zelfs tot in de kleine uurtjes.

Yannis Haralambous besprak vervolgens 'Omega, the full release' en Michel Goossens vertelde over de bekende 'T_EX Live 2' cdrom en enkele belangwekkende aspecten van 'Web2c' en 'L^AT_EX catalogues'.

Werner Lemberg introduceerde enkele nieuwe programma's om fonts te manipuleren, en Hans Hagen sloot de middag af met een lezing over het maken van complexe en hoog-interactieve documenten.

De volgende dag werd dieper ingegaan op zaken rond elektronisch publiceren. Elsevier heeft Yannis Haralambous gevraagd een nieuwe set mathematische fonts à la Times te maken. Yannis liet enkele details zien van Griekse symbolen die hij had ontworpen. Sergey Lesenko vertelde over zijn dvi/pdf programma en het gebruik van graphics. Han The Than liet vervolgens zien hoe je direct PDF kan produceren met 'PDF_TE_X'. Ontwikkelingen in PDF en L^AT_EX waren het thema van de lezing van T.V. Raman van Adobe Systems. Ook IBM was van de partij. Hun vertegenwoordiger vertelde over 'techexplorer', een programma waarmee (L^A)T_EX-code *on-the-fly* geformatteerd kan wor-

den zoals we dat gewend zijn van WWW-browsers.

In de middag besprak Jon Bosak ‘DSSSL’, voor SGML-kenners een begrip. SGML is zoals bekend de ‘grote broer’ van HTML. Binnenkort komt XML er aan, de opvolger (‘grotere broer’) van HTML. Dat wordt vast heel bijzonder.

Sebastian Rathz liet zien hoe \TeX als *backend* voor ‘Jade’ kan functioneren. Ross Moore ging vervolgens in op heden, verleden en toekomst van ‘ \LaTeX 2html’. Daarna demonstreerde Eitan Gurari een radicaal andere benadering zien om van *any* \TeX naar HTML te converteren, namelijk via DVI, met behulp van het pakket ‘ \TeX 4ht’.

Op woensdagochtend kwamen de uitgevers aan bod. Mimi Jett vertelde als hoofd van een grote uitgeverij over haar ervaringen met \LaTeX bij de productie van computer ondersteund onderwijs. Daarna liet Douglas Lovell een onverwachte toepassing van \TeX zien, namelijk voor het automatisch genereren van autohuurcontracten. Fred Bartlett ging vervolgens in op de eisen die uitgevers stellen aan \LaTeX om het beter bruikbaar te maken voor hen. Met een (mijns inziens nogal lauwe) panel discussie werd de ochtend afgesloten.

In de middag kwam \LaTeX 3 aan bod in een lezing van David Arlisle. Verder liet Michael Downes zien dat het breken van formules over meerdere regels weliswaar lastig is, maar met wat handigheid toch grotendeels automatisch netjes kan gebeuren. Ook de middag werd afgesloten met een panel discussie, met als thema “intwikkelingen in \LaTeX in relatie tot het verschijnen van nieuwe \TeX -varianten”. Men was nogal moe geloof ik.

De laatste dag was gewijd aan meer exotische zaken zoals kleurgebruik en tekst in minder eenvoudig te zetten talen (althans vanuit \TeX -perspectief). Dan Olson beschreef hoe hij met vallen en opstaan de kneepjes van het afdrukken in kleur onder de knie kreeg. Kazuhiro Kitagawa vertelde over zijn onderzoeksproject naar de mogelijkheden om \TeX te gebruiken bij het maken van watermerken.

’s Middags vertelde Frank Mittelbach over wat er allemaal komt kijken bij het maken van multi-linguale documenten. Hans Hagen liet daarna zien hoe in ‘Con \TeX t’ met *interfaces* in verschillende talen kan worden gewerkt. De laatste lezing was van Werner Lemberg die liet zien hoe met het \LaTeX -pakket ‘CJK’ Chinese, Japanse en Koreaanse teksten gezet kunnen worden. Hij liet daarbij ook zien hoe fraai al die voor ons zo vreemde tekens gelijktijdig zichtbaar kunnen worden gemaakt en ge-edit(!) met behulp van Mule.

Prijzen

De prijs voor “het beste / meest interessante” ging dit jaar naar Han The Than voor zijn werk aan PDF \TeX . Hij verdient die prijs zonder meer. Toch is het vreemd dat hij die prijs kreeg omdat vorig jaar op de TUG-conferentie in Dubna (Rusland) Petr Sojka voor hetzelfde werk al onderscheiden is. Maar daar waren vrijwel geen Amerikanen bij...

Michel Goossens werd als scheidend voorzitter onderscheiden voor zijn vele verdiensten. Mimi Jett volgt hem op. Zij heeft er duidelijk zin in en we wensen haar veel succes. Ook van Sebastian Rathz werd op passende wijze afscheid genomen omdat hij aftrad als secretaris.

Uiteraard werden ook verschillende anderen die voor of achter de schermen mee hadden geholpen aan het realiseren van deze conferentie bedankt.

Verder werd Hans Hagen onderscheiden als meest opvallende / veelbelovende nieuwkomer in de TUG-wereld.

Social Event

Bij een TUG-conferentie hoort een “Social Event”. Dit keer was dat een rondvaart met diner door de baai van San Francisco. Bij ondergaande zon konden we genieten van uitstekend eten en drinken en uitzicht op markante punten zoals de fameuze Golden Gate Bridge en het gevangenis-eiland Elcatraz.

Hier kwam \TeX echt terug bij zijn “roots”: Donald Knuth was ook present op de boottocht.

Conclusie

Wat mij betreft was het een heel geslaagde bijeenkomst. Zoals gewoonlijk werd er in de wandelgangen uitvoerig gediscussieerd over allerlei \TeX nische zaken, vaak tot in de kleine uurtjes. Er werden banden gesmeed en aangehaald. Vele deelnemers zijn van naam erg bekend, maar ze een keer *zien* en met ze praten is toch heel anders dan communiceren via minder persoonlijke media zoals elektronische post.

De volgende TUG-conferentie zal plaatsvinden bij de Nicholas Copernicus Universiteit in Toruń in Polen. Het thema zal zijn: “Integrating \TeX with the surrounding world”. Dat wordt vast ook een interessante bijeenkomst. Bovendien is die voor ons Europeanen aanzienlijk dichterbij (lees: goedkoper) dan in San Francisco. Van harte aanbevolen.

Bijlage 8

Installatie-handleidinkje voor 'L^AT_EX voor Windows 1.0'

Robbert Hardin

abstract

Op verzoek van Frans Goddijn heeft FGBBS-user Robbert Hardin een verslag geschreven van de installatie van het pakket L^AT_EX voor Windows (16 diskettes, ook op FGBBS te vinden). Disk 10 die in beschadigde versie op het bbs bleek te staan, is door Robbert vervangen door een nieuw, gaaf exemplaar.

keywords

L^AT_EX, Microsoft Windows, systems, installation

Voordat je gaat proberen 'L^AT_EX voor Windows 1.0' te installeren is het handig om te weten wat de systeemeisen zijn:

- IBM-Pc of compatibel,
- MS-Dos 6.0 (of beter) als besturingssysteem en Windows 3.1 (of beter) als shell (er zijn reeds succesvolle installaties op Windows 95 en Windows NT gesignaleerd),
- 80386-processor met co-processor (of beter),
- 45 MB op de (een) harde schijf,
- minstens 4 MB RAM.

'L^AT_EX voor Windows 1.0' bestaat uit vier delen:

- WinEdit,
- GhostScript/GSView,
- LaTeX2_ε,
- ondersteunende bestanden/programma's

WinEdit is een niet-geregistreerde shareware-versie van WinEdit (versie 96v), een Windows-editor waarmee je ASCII-bestanden mee kunt aanmaken. WinEdit is op L^AT_EX afgestemd, waardoor het handiger is dan een willekeurige ASCII-editor als Edlin.

GhostScript/GSView is een PostScript-previewer bestaande uit Aladdin GhostScript (versie 4.03) en GSView32 (versie 2.1).

LaTeX2_ε bevat de volledige Dos-versie van T_EX, emT_EX. De DOS-componenten worden echter niet gebruikt; in plaats daarvan worden Windows-componenten gebruikt. Tevens bevat het het macropakket L^AT_EX 2_ε (versie januari 1997). Voor L^AT_EX 2_ε zijn er geen omgevings-

variabelen (zoals EMTEXDIR, DVIDRVINPUT, DVIDRV-FONTS) nodig, zij dienen zelfs verwijderd te worden.

De ondersteunende bestanden bevatten (algemene) informatie (Tree.Txt, 00readme, Message, Bugfix.Txt) en een Local Guide (Bookeven.Ps, Bookodd.Ps, Local.Ps). De ondersteunende programma's zijn een unzipper (Pkunzip.Exe) en een programma dat 32-bits programma's onder Windows ondersteunt. Dit laatste programma is nodig omdat GhostScript/GSView 32-bits programma's zijn.

Na deze inleiding hebben we genoeg achtergrondkennis om 'L^AT_EX voor Windows 1.0' te installeren.

1 Voorbereidingen

1.1 Verwijder oude omgevingsvariabelen en oude T_EX-versies

'L^AT_EX voor Windows 1.0' heeft geen omgevingsvariabelen nodig. Omgevingsvariabelen, zoals EMTEXDIR, DVIDRVINPUT en DVIDRVFONTS moeten verwijderd worden uit \Autoexec.Bat en \Config.Sys. Oude T_EX-versies moeten verwijderd of hernoemd worden.

1.2 Installatie 32-bits ondersteuning

Omdat GhostScript/GSView 32-bits programma's zijn moet Windows 32-bits ondersteuning geven. Gebruikers wier Windows versie geen 32-bits ondersteuning geeft moeten het programma Pw1118.Exe executeren. Dit is een zelf-uitpakkend programma dat W32s125 (een 32-bits ondersteuningsprogramma voor Windows) installeert. Tevens wordt het 32-bits programma Freecel geïnstalleerd. Als Freecel werkt dan is de installatie gelukt.

1.3 Uitpakken .Zip-files

De bestanden zijn dubbel gezipd. Daar bedoel ik mee dat de bestanden gezipd zijn, vervolgens zijn ze in groepjes van 1,44 MB verdeeld en deze groepjes zijn ten slotte op schijf gezipd (onder de naam Disk01.Zip tot en met Disk16.Zip). Het installatieprogramma werkt met de enkel-gezipde bestanden. Dat betekent dat je Disk01.Zip tot en met Disk16.Zip moet unzippen. Unzippen kan met het meegeleverde Pkunzip.Exe. Het installatieprogramma geeft je de keuze tussen installeren vanaf diskette en vanaf harddisk.

Wie vanaf diskette wil installeren zal dus naar diskette moeten unzippen. Dit kan met Pkunzip Disk01 a:. Wie

vanaf harddisk wil installeren zal dus naar harddisk moeten unzippen. Dit mag op twee manieren:

1. Unzip alles naar een directory; `\winlatex` bij voorbeeld. Dit kan met `Pkunzip Disk01 \winlatex`.
2. Unzip elke disk naar zijn eigen subdirectory; `\winlatex\Disk01`, bij voorbeeld. Dit kan met `Pkunzip Disk01 \winlatex\Disk01`.

LET OP: `Disk02.Zip` bevat `WE96v16i.Zip` en `WE96v32i.Zip` dit zijn resp. de 16-bits versie en de 32-bits versie van WinEdit. Je kan er echter maar een gebruiken. Welke je kunt gebruiken wordt tijdens het installeren duidelijk. Wie vanaf diskette wil installeren zal `WE96v16i.Zip` en `WE96v32i.Zip` naar diskettes moeten kopiëren (dus NIET unzippen) en tijdens de installatie de juiste diskette in de computer stoppen. Wie vanaf harddisk wil installeren zal `WE96v16i.Zip` en `WE96v32i.Zip` naar de harddisk moeten kopiëren (dus NIET unzippen). Als je methode 1. gebruikt moet je `WE96v16i.Zip` en `WE96v32i.Zip` naar `\winlatex` kopiëren. Als je methode 2. gebruikt moet je `WE96v16i.Zip` en `WE96v32i.Zip` naar `\winlatex\Disk02` copieren. Tijdens de installatie wordt dan de goede versie van WinEdit geïnstalleerd.

1.4 Verzet de kalender

WinEdit is een niet-geregistreerde shareware-versie van WinEdit (versie 96v). Wanneer dit programma na 21 dagen niet van een licentie is voorzien zal het zichzelf verwijderen. Door de datum op 01 december 1999 te zetten krijg je wat langer de tijd om WinEdit te testen en te beslissen of het aan je eisen/verwachtingen voldoet. Als je dacht dat je aan 21 dagen genoeg had, maar je wilt wat extra bedenktijd, dan kun je de bedenktijd verlengen door de installatiedatum in `WinEdit.Log` en `WinEdit.Bak` in de directory `\emTeX\WinEdit` aan te passen. Overigens raad ik registratie van harte aan. WinEdit is een handig programma en niet alleen voor \TeX -doeleinden.

2 De installatie

2.1 Start

Start Windows. Executeer `TeXinst.Exe`. Dit bestand bevindt zich in `Disk01`. De installatie wordt nu gestart. Het is niet mogelijk de installatie af te breken. Wel is het mogelijk de installatie te onderbreken.

2.2 Installatie WinEdit

WinEdit wordt geïnstalleerd met behulp van het originele installatiescript van WinEdit. Als WinEdit iets vraagt, antwoord dan steeds met ja, ik wil. Op de vraag waar WinEdit geïnstalleerd moet worden suggereert WinEdit `C:\emTeX\WinEdit`. Het is mogelijk hier een andere di-

rectory te kiezen, aanbevelswaardig is dat echter niet. Het kiezen van een andere harddisk zou geen problemen moeten opleveren. Let er goed op dat je de juiste WinEdit installeert; het feit dat je Windows nu 32-bits programma's ondersteunt betekent niet dat je de 32-bit versie van WinEdit kunt gebruiken! Tijdens de installatie worden sommige WinEdit-menu's aangepast. Het is mogelijk om de installatie te onderbreken om een backup te maken van de originele WinEdit-menu's.

2.3 Installatie GhostScript/GSView

GhostScript/GSView wordt geïnstalleerd met behulp van het originele installatiescript van GhostScript/GSView. Als GhostScript/GSView iets vraagt, antwoord dan steeds met ja, ik wil. Op de vraag waar GhostScript/GSView geïnstalleerd moet worden MOET je dezelfde harddisk kiezen als je voor WinEdit gekozen hebt! Wanneer je GhostScript/GSView voor de eerste keer oproept moet je een hele vragenlijst afwerken. Beantwoord alle vragen met ja, ik wil. Wanneer GhostScript/GSView geïnstalleerd is kun je de Local Guide uitprinten; dubbelklik op `GSView`, open het ondersteunende bestand `Local.Ps` en druk op p om het printmenu op te roepen.

2.4 Installatie van LaTeX2_ε

Vragen van $\LaTeX 2_{\epsilon}$ dienen naar eer en geweten beantwoord te worden.

3 Afwerking

3.1 Verzet de kalender

Zet eventueel de kalender gelijk.

3.2 Bugs

- `Disk10` is verkeerd gezippt. Een goede versie is te verkrijgen bij `ftp.minet.uni-jena.de` in de directory `(\Pub\TeX)`.
- Volgens de Local Guide (2.4) heb je na de installatie een directory `\Gstools` op C:. Ik heb hem echter op D:, gelukkig heeft \LaTeX daar geen moeite mee.
- Geen echte bug, maar wel vervelend; het macropakket Babel ontbreekt (of ik heb niet goed gekeken).

3.3 Wat nu?

Het is handig om ' \LaTeX voor Windows 1.0' te testen:

- Open WinEdit
- Open `Testpage.Tex` in `\emTeX\TeXinput\LaTeX2e`
- Druk op F6, nu wordt er een `.Dvi`-bestand gemaakt
- Druk op F7, hiermee start je de PostScript-previewer
- Keer terug naar WinEdit en druk op F9, nu wordt het printvenster geopend
- Print `Testpage` en stel eventueel je printer bij

Tot Slot

Deze installatie-handleiding is verre van volledig. Als ik hem echter had gehad toen ik 'L^AT_EX voor Windows 1.0' ging installeren dan was de installatie een stuk vlotter verlopen. Ik herinner me echter nog een eerdere poging van mij om emTeX te installeren. Mij hoor je daarom niet klagen. Alles wat in dit handleidinkje staat is afkomstig van de Local Guide en mijn installatiepogingen. Mijn advies is om vooral 2.1, 2.3, 2.4 en 8 te lezen.

Succes

Bijlage 9

L^AT_EX & Chinees

Erik Frambach
Rijksuniversiteit Groningen
email: E.H.M.Frambach@eco.rug.nl

abstract

Met behulp van het L^AT_EX-package ‘CJK’ is het mogelijk om tekst te zetten in het Chinees, Japans en Koreaans. In dit artikel laat ik in het kort zien wat ervoor nodig is om Chinese tekst met L^AT_EX — Japans en Koreaans blijven buiten beschouwing. Het installeren van de benodigde software wordt behandeld, evenals het gebruik van dit systeem in de praktijk.

keywords

L^AT_EX, Chinees, CJK, fonts, PostScript, Big 5

Introductie

Chinees is voor computers een lastige taal omdat het een enorm groot karakters omvat, in de orde van grootte van **10.000** tot **50.000**. Een toetsenbord met alle karakters, zoals we met westerse talen gewend zijn, is daarom ondoenlijk.

Maar niet alleen het invoeren van Chinese tekst is daarvoor lastiger, ook voor het opslaan moeten andere afspraken gemaakt worden. Het is duidelijk dat een Chinese tekst niet kan worden gerepresenteerd met 1 byte per teken, zoals bij voorbeeld bij Nederlands. Daarvoor zijn minstens 2 bytes nodig. In de loop van de tijd zijn daarvoor verschillende coderingen bedacht, zoals *BG5*, *Big-5* en *CNF*.

Voor het invoeren bestaan inmiddels redelijk gebruikersvriendelijke programma’s die teksten in verschillende coderingen kunnen im- en exporteren.

CJK

Het L^AT_EX-package ‘CJK’ van Werner Lemberg implementeert een vertaalslag van o.a. BG5- en Big-5-gecodeerde teksten naar T_EX-fonts. Daarnaast biedt CJK een heleboel extra’s die nodig zijn om serieus tekst te kunnen zetten in het Chinees, wat verder gaat dan een enkele kreet. CJK definieert bij voorbeeld teksten als ‘Samenvatting’ en ‘Inhoud’ in het Chinees, maar zorgt er ook voor dat ‘Hoofdstuk 1’ op de Chinese manier wordt geschreven: ‘Eerste hoofdstuk’.

Verder zitten er aardigheden in CJK zoals ‘pinyin’, een transcriptie van Chinees naar Latijns alfabet met accen-

ten, en ‘ruby’s’, kleine hulpjes bij de uitspraak van minder gangbare karakters.

Chinees T_EX-fonts

Een minstens zo belangrijk probleem dat CJK oplost is de vertaling naar T_EX-fonts. T_EX kan in een font maximaal **255** karakters opslaan. Daarom moet voor een compleet Chinees fonts de zaak in stukken worden geknipt. CJK heeft dat zo efficiënt mogelijk gedaan, waardoor maximaal ‘slechts’ **55** (sub)fonts nodig zijn. Om het ook voor T_EX behapbaar te maken worden fonts pas geladen als dat echt nodig is. In de praktijk blijkt dat veel teksten met een stuk of tien (sub)fonts nodig zijn. Intern zijn de fontmetrieken overigens erg simpel. Chinees is niet-proportioneel, dus alle “boxen” zijn even breed.

HBF, TrueType en PostScript

Qua fontmetrieken is het dus eenvoudig, maar qua rendering niet. In principe geeft CJK drie mogelijkheden:

- **hbf2gf**: een zogenaamd HBF-font kan met een eenvoudig programma naar ‘gf’ vertaald worden. ‘gf’ kan op zijn beurt weer naar ‘pk’ vertaald worden, een formaat waar elke div-driver mee overweg kan.
- **ttf2pk**: een TrueType-font kan met een eenvoudig programma direct naar ‘pk’ vertaald worden.
- **ps2pk**: idem maar dan met een PostScript-font.

Chinees HBF-, TrueType- en PostScript-fonts zijn hier en daar op Internet te vinden, maar veel is er niet. Ook zijn er verschillen in de compleetheid van fonts. Sommige fonts kunnen “vrij” gebruikt worden, andere alleen voor niet-commerciële toepassingen.

Verder is er een conversieprogramma (een contributie aan CJK) om volledige TrueType-fonts (dus met *alle* karakters!) in kleinere PostScript subfonts te vertalen. Eén zo’n grote TTF (4 tot 8 megabyte) geeft dan 55 PostScript Type 1 fonts van 100 tot 200 kilobyte per stuk.

Omdat **ps2pk** al volledig geïntegreerd is in 4T_EX heb ik ervoor gekozen via die weg pk-fonts te genereren. Als PostScript als uitvoer wordt gekozen worden direct de PostScript Type 1 fonts gebruikt.

Bij dat laatste moet aangetekend worden dat *partial font downloading* daarvoor van groot belang is. Veel PostScript-printers verslikken zich in de massa fonts die

voor een eenvoudig stuk al gauw nodig zijn. Bovendien wordt uit elk font vaak maar een erg klein deel daadwerkelijk gebruikt, zodat de print file ook onevenredig groot zou worden. Gelukkig ondersteunt DVIPS tegenwoordig partial font downloading niet alleen voor PK-fonts, maar ook voor Type 1 fonts. Een dvi-drivers als em \TeX 's DVIHPLJ stuurt ook alleen de gebruikte karakters naar de printer.

Er is een 8-tal min of meer volledige Chinese TrueType-fonts op Internet te vinden die ik naar PostScript subfonts heb vertaald (kost ca. 10 minuten per stuk op een Pentium 166). Bij die vertaling worden 8×55 Type 1 fonts gegenereerd, naast 8×55 tfm-bestanden. Als al die bestanden op de juiste plek zijn geïnstalleerd hoeft alleen hier en daar nog een \LaTeX -fontdefinitie aangepast te worden in de CJK \TeX input directory.

Van Big-5 naar CJK

Voordat we daadwerkelijk met \LaTeX en CJK aan de slag gaan moeten we eerst de Big-5 tekst bewerken. In de code kunnen namelijk backslashes en tekens zoals '{' en '}' voorkomen die, zoals bekend, in \TeX een bijzondere betekenis hebben.

Een conversieprogramma wordt daarvoor meegeleverd. Dat programma schrijft vooraan het bestand een regeltje zodat CJK weet dat het bestand voorberekt is, en het vangt alle 2-byte karakters door een iets andere representatie die niet met \TeX interfereert. Terugvertaling naar het origineel wordt niet ondersteund, dus maak backups als je de tekst later nog met een Chinese editor verder wilt bewerken.

CJK in de praktijk

Als voorbeeld nemen we een klein stukje Chinese tekst dat met CJK wordt meegeleverd:

```
% CJK example
\documentclass[12pt]{article}
\usepackage{CJK}
\begin{document}
\begin{CJK*}{Bg5}{kai}
\CJKtilde
... onleesbare tekens...\textrm{Big-5}~...
onleesbare tekens ... et cetera...
\end{CJK*}
\end{document}
```

Merk op dat binnen een stuk Chinese tekst gewoon \TeX -commando's uitgevoerd kunnen worden, bv. omschakelen naar romeins of vet.

De tekst in dit bestand is puur Big-5 en moet dus nog voorberekt worden voor CJK. Na voorbereking ziet het er zo uit:

```
\def\CJKpreproc{}% CJK example
\documentclass[12pt]{article}
\usepackage{CJK}
\begin{document}
\begin{CJK*}{Bg5}{ka}
\CJKtilde
...onleesbaar teken 187onleesbaar teken 229...
onleesbaar teken 68... ~\textrm{Big-5} ...
onleesbaar teken 243 ...
\end{CJK*}
\end{document}
```

De precieze betekenis van commando's die CJK allemaal definieert ga ik hier hier uitleggen, daarvoor is een uitgebreide handleiding beschikbaar. De meest basale vermeld ik hieronder.

$\begin{CJK*}{Bg5}{ka}$

Met dit command begin je een \LaTeX -environment waarin tekst staat die Big-5-gecodeerd is. Het font dat gebruikt wordt is 'ka'. Dat betekent dat er een fontdefinitiebestand c00ka.fd moet zijn.

Zo'n fontdefinitiebestand ziet er bij voorbeeld zo uit:

```
\def\fileversion{4.1.3}
\def\filedate{1997/06/20}
\ProvidesFile{c00ka.fd}%
  [\filedate\space\fileversion]
\DeclareFontFamily{C00}{ka}{}
\DeclareFontShape{C00}{ka}{m}{n}%
  {<-> CJK * b5ka}{}
\DeclareFontShape{C00}{ka}{bx}{n}%
  {<-> CJKb * b5ka}{\CJKbold}
```

\LaTeX /CJK verwacht dat de fonts b5ka01 t/m b5ka55 beschikbaar zijn, dat wil zeggen tfm-bestanden. Voor uitvoer zijn overeenkomstige pk- of pfb-bestanden nodig.

Het verschil tussen de CJK en de CJK* omgeving is dat in de laatste spaties en regelovergangen worden genegeerd. Meestal is dat wat je wil. Merk op dat regels niet op willekeurige plekken afgebroken kunnen worden, omdat elk Chinees karakter 2 tekens inneemt. Midden in een karakter breken zal niet goed gaan.

\CJKbold

CJK emuleert desgewenst een vette variant van een willekeurig CJK-font door karakters meerdere keren te printen met heel kleine verschuivingen. Het Chinees kent eigenlijk geen vet of cursief zoals wij dat kennen, maar voor sommige toepassingen is deze methode heel handig. In een stuk Chinese tekst kan nu op de gewone manier (met $\backslash bf$ of $\backslash textbf$) vet geselecteerd worden. Cursief is er niet.

Overigens werken `\small`, `\Large` en dergelijke ook gewoon.

`\CJKcaption`

Met dit commando kan een ‘caption’ bestand geladen worden. Daarmee worden de juiste definities geladen voor zaken als ‘Inhoud’ en ‘Samenvatting’. Voorbeeld: `\CJKcaption{Bg5}`. Eigenlijk is dit nog niet genoeg om Chinese tekst op typografisch verantwoorde wijze te zetten. Om ‘Eerste hoofdstuk’ en dergelijke te krijgen is meer nodig. Met behulp van het \LaTeX -pakket `koma-script` is dit te realiseren. In Werner Lembergs artikel in *Die \TeX nische Komödie* wordt in detail uitgelegd hoe dat moet. In ongeveer tien regels aan code is dat te implementeren.

Pinyin

CJK definieert ook de ‘pinyin’ stijl. Dat is een soort transcriptiemethode voor Chinees naar westerse tekens. Het \LaTeX -package heet `pinyin.sty` maar is, in tegenstelling tot de kern van CJK, niet afhankelijk van \LaTeX : in plain \TeX kun je de stijl ook gebruiken door het commando `\input pinyin.sty` te geven.

Een stukje pinyin, met ‘ruby’s’ erboven, ziet er bij voorbeeld zo uit:

我 很 喜 歡 吃 中 國 飯 。
wǒ hěn xǐhuān chī Zhōngguó fàn .

Dit betekent: ik eet graag Chinees. De \TeX -code voor pinyin is voor iedereen die iets van Chinees weet heel voor de hand liggend: `\Wo3 \hen3 \xi3\huan1 \chi1 \Zhong1\guo2 \fan4`

Verticaal schrift

Traditioneel Chinees schrift loopt in kolommen van boven naar onder. De kolommen lopen van rechts naar links. CJK heeft ook hiervoor een oplossing die voor eenvoudige gevallen goed genoeg is: domweg alle karakters een kwart slag draaien! De papieren afdruk draai je vervolgens een kwartslag terug et voilà! Typografen zullen echter snel fouten kunnen aanwijzen, met name bij de interpunctie die in verticale tekst anders kan zijn dan in horizontale.

Voorbeelden

Nu we de belangrijkste kenmerken gezien hebben laat ik wat stukjes zien uit de in het Chinees in \LaTeX geschreven handleiding van CJK.

De titelpagina van de handleiding:

Big-5 碼中文系統下的 CJK \LaTeX 2_ε 排版系統 (For CJK version 4.1.0)

李君宇 (Chun-Yu Lee)

Werner Lemberg

Een stukje uit sectie 5, over het installeren van fonts:

5 字型安裝

本文僅以由 True Type 字型產生的 PK 字型或 PS 字型為說明。字型之取得參考附錄 B 之說明。

PK 字型 (或 PS 字型) 最好安裝在當你 要升級 \TeX / \LaTeX 系統時不會影響的位置, 因為一旦此類字型產生後, 即可長時期的使用; 另外, 如果每次要升級 CJK 系統或 \TeX / \LaTeX 系統時, 也可放心大膽的下手。

Hetzelfde stukje, maar nu niet met het ‘KaiSu’ font maar met ‘ChuYuanTi’:

5 字型安裝

本文僅以由 True Type 字型產生的 PK 字型或 PS 字型為說明。字型之取得參考附錄 B 之說明。

PK 字型 (或 PS 字型) 最好安裝在當你 要升級 \TeX / \LaTeX 系統時不會影響的位置, 因為一旦此類字型產生後, 即可長時期的使用; 另外, 如果每次要升級 CJK 系統或 \TeX / \LaTeX 系統時, 也可放心大膽的下手。

Meer weten

Het CJK-pakket is op CTAN (Comprehensive \TeX Archive Network) te vinden in de directory `language/chinese/-CJK`. Een aantal fonts zijn te vinden in `fonts/CJK`. Op de 4all \TeX cdroms versie 4 is het pakket (althans het Chinese deel) volledig geïnstalleerd.

Als achtergrond is het verstandig te lezen “Das CJK-Paket für \LaTeX 2_ε” van Werner Lemberg, in *Die \TeX nische Komödie* 4/1995. In de nog te verschijnen proceedings van de TUG conferentie 1997 zal een nieuwer en uitgebreider artikel over CJK in het Engels verschijnen.

T_EX inside, insight, in sight: get priorities right

Kees van der Laan

abstract

It is argued that for using T_EX – a multi-level tool – T_EX inside knowledge should not be necessary for the layman, for production purposes. The ‘why, what and when,’ especially at the grey level will be discussed. Necessary insight issues are enumerated.

keywords

Active list separator, education, greyboxes, hyphenation accented words, macrowriting, minimal markup, partitioned matrices, two-part macros.

1 Introduction

We all love T_EX, that is those people who gather at user meetings. But, . . . apparently – and the theme of the GUST 97 meeting witnesses it, as was the earlier TUG slogan ‘develop your T_EXpertise’ – there is a need for knowledge of T_EX’s inner workings. In general I don’t agree.¹ Wizards do need it, but what does the layman?

T_EX inside, or . . . insight in T_EX?

Below material has been borrowed from my earlier Paradigm notes, and some more.

1.1 Levels of understanding

T_EX, and its user interfaces, can be seen as a well-documented multi-level tool for EP, and communication in general. Its quality, general availability, being for free, openness, and extensibility are necessary issues which contributed to its success. Its abstraction from input and output devices are genuine gems.

Are there any disadvantages? For daily use a tool should be simple,² should be of the push-the-button type to a high degree. Knuth envisioned even that users should create their own blend of T_EX for any significant application, adapted to the purpose and therefore simple in use.³

What I did in BLUE T_EX was precisely in Knuth’s spirit of personalizing T_EX, though not at the level Knuth foresaw. I did it on the macro level, by the way like Knuth with his manmac – the macro collection he used for typesetting The T_EXbook and The METAFONTbook – and like all other writers of macro collections.⁴ I abstracted from the

microscopics, from the complexity, in abstracting towards macroscopic commands, in the push-the-button spirit, such that I could forget about most of the details of T_EX, except from some general knowledge. Similarly to Knuth, and differently from most other macrowriters, I adopted the minimal markup paradigm.⁵

Copy can best be setup just in ASCII with visual layout, and only when finished with the contents a few markup commands need to be inserted, et voilà you will end up with a beautiful typeset result, at the expense of little markup, and as nearly an unblurred script as possible.⁶

Example (From the PWT guide)

Let us assume that we have ordinary text, to be typeset by computer. What to do? Let us begin by the beginning and write it down.

An unexpected party

In a hole in the ground there lived a hobbit.
Not a nasty, dirty, wet hole filled with the
ends of worms and an oozy smell,
nor yet a dry, bare sandy hole with nothing
in it to sit down on or to eat:
it was a hobbit-hole and that means comfort.

1. It is similar to driving cars. A little nodding knowledge is beneficial. Inside knowledge was only necessary in the beginning.
2. T_EX, more honestly EP, is not simple. It forms a complex, captivating, and challenging field.
3. What happened was that T_EX has been ported to many if not all conceivable platforms. METAFONT has been adapted for graphical purposes as extra to font generation in Knuth’s spirit by John Hobby in his MetaPost.
4. ϵ -T_EX is different. More in the spirit of what Knuth foresaw. It added primitives next to the processing modes: compatible, extendible, enhanceable. However, it is not driven by a concrete, huge, boundary-challenging task. Therefore, I can’t tell the difference from ‘creeping featurism.’ Nothing wrong with that, IMHO, with all respect. It bypasses, however, the personalizing aspect of T_EX, that is many a T_EXie personalizes T_EX in order to fulfill his/her not-so-huge nor boundary-challenging tasks, towards a *simpler language*, abstracting from details, eventually in the jargon of his/her trade. A much overlooked aspect IMHO.
5. By this I mean that no superfluous (markup) symbols should be necessary – as an example think of the curly braces mania – nor should concepts alien to the field (EP) be introduced. It should be natural and behave the way you expect it to behave, in short T_EX should be customized.
6. I agree that this is an oversimplification. In reality I directly start from an empty template for my notes.

It had a perfectly round door like a porthole, painted green,...

We must instruct the computer – that is, supply markup – to make something nice out of it. How?

As far as I understand it, Knuth’s markup comes down to just preceding the title by `\beginchapter`. That is all. He makes use of the already available visual markup of blank lines. These blank lines separate document parts, be it a title from the text or paragraphs from each other.

Is that all? Essentially, yes. But – there is always a but – the role which `\beginchapter` has to play is not simple. Happily, that is at this moment of use not our concern. That is the job of the book designer in cooperation with the programmer, or if we did it ourselves some activity from the past.

I take for granted the niceties of \TeX ’s automatically text processing features – and which a regular user tends to forget – such as automatic handling of ligatures, justification and hyphenation over paragraphs, and ignoring of superfluous spaces in text. All this is not necessarily restricted to English. Again, nodding knowledge is beneficial.

Only after many a year I realized that the best way of using \TeX is not to use it to start with. Just ASCII with visual layout, with as top-priority to get the contents right.

Once we are able to achieve this, we may continue, because it’s of no use to have a beautiful typeset document of which the contents is rubbish, trivial, and in general not worth to be communicated.

To make a long story short: I abstracted from the tool to such an extend that I could forget about most of it.

1.2 Audience: the user life-cycle

Users are alive and tend to move: they come, develop and go. I presume they will end up as knowledgeable users, who ... will forget about most of the details of the tool. They have lost interest, are burnt out.

1.3 Script: minimal markup

The result of the above attitude is that the markup of the script at the outer level is simple and equivalent to

SGML modulo *<some syntactic sugar>*

I not only like the typeset results but also the elegance of \TeX markup.

Another benefit comes with reuse. My main reuse is to convert my `BLUe` \TeX scripts conforming to MAPS style, that is into \LaTeX . For the outer level markup I use my convertor-assistant written in \TeX , which replace `\bluehead...` into `\section{...}` and so on; no retyping.⁷

Lower level markup tags – (complex) tables, math, graphics and so on – are left invariant and I include the cor-

responding macros along with the converted script. This way of working is similar to handling `POSTSCRIPT` graphics. There too the invokes are left invariant and the converted script is accompanied by the `POSTSCRIPT` files.

2 \TeX as a blackbox?

Can \TeX be used as such? Alas, history has it, that (computer-assisted) typesetting is too complex a job. We need greybox awareness, at best by customizing \TeX first towards your personal needs. IMHO, we need awareness of what is going on and how to discern our wishes into realistic ones.

Example (*Verbatim in headings*)

In this (\TeX nical) note I needed a control sequence as part of the heading. The backslash was needed. How to do that? At the expense of another font – kind and size – in the heading a non-context sensitive solution is the use of `\cs`.

```
\blueexample \cs{begindisplay}...
```

It appears correctly in the heading and in the ToC. With `\blueexample` and ilks the in-line verbatim does not work, even worse, horrible error messages come your way. `BLUe` is not robust – as is the same with any personalized format, I guess – in the sense that you can use anything anywhere is not yet in sight, alas. Moreover, its size should by default kameleon.

3 \TeX insight: greyboxes

I grew up with the paradigm of ‘greyboxes,’ especially in numerical mathematics. It was considered impractical as a user to lose oneself in the details. To understand roughly the ideas underlying the algorithm, its virtues and limitations was enough for using a library routine. Take for example the infamous gem of zerofinding as implemented in the zerofinder `zeroin`. The complexity of the mixture of strategies is not necessary to be known by the user. For a user the invariance: a shrinking interval with the function values at the endpoints of opposite sign, is all that has to be known. After completion the interval is smaller than the requested precision. In other words a small interval will result on which the function changes sign. For a continuous function this implies that a zero of the function has been located within the required precision. Awareness that convergence is superlinear does not harm.

7. I must confess, however, that it has developed little as yet, so it is a personal tool. Of course, one can use Perl or similar scripting tools, but I liked to exercise \TeX for this, with as result that it is sufficient for my purpose.

3.1 Why greyboxes in T_EX?

Macrowriting – restricted to creating your push-the-button personalized level of T_EX – needs greyboxes awareness.

Is it handy to know more, to understand T_EX the program?

I don't know, I guess it should be handy, but at what price, what is the investment versus the benefits? I could do without, which saved me a lot of time and ... mental luggage. For that reason I consider myself as not to belong to the T_EX-wizards clan, if ever I would like to belong to a clan.

Basically, I needed enough knowledge to be able to read and understand the already available macro collections. And I must say that a few of these collections are just too clever, too complicated and too huge. A lot of dependencies not in the least due to metaness – misplaced IMHO, with all respect – and they suffer from unduly general philosophies, while a pragmatic here-and-now approach would have served the purpose, not in the least the maintenance issue, *casu quo* the adaptability to change in general.⁸

3.2 What greyboxes knowledge of T_EX?

Necessary is

- tokens are characterized by ASCII-catcode pair
- I/O abstraction
- the boxes, glue and penalties approach
- linebreaking works over paragraphs (how paragraph parameters work)
- how copy is reused (stored within T_EX or within a file)
- awareness of (rigid) discrete sized fonts sets.

For macrowriters the following extras

- processing on-the-fly via two-part macros
- the concept of the active list separator
- the use of token variables for options.

Let me give you an example, the macro to typeset a heading, and to reuse the title in the toc and index. Maybe all the above knowledge is not necessary if we start from templates, for example the `\beginsection` macro as supplied by Knuth. The toc and index requirements introduce a lot of extra difficulties. Let us first concentrate on the limited problem and impose ourselves some restrictions. Why not go for simplicity first?

3.3 Heading

First of all we should clearly state the design goals, which activity by the way is much neglected in the T_EX world. Much is done on-the-fly.

Design goals From a typographer's point of view I would like for the typesetting of the heading to

- discourage to set the title alone at the end of a page
- typeset (flexible) vertical space before (big) and after (med) the title
- gobble spaces at the beginning of the title
- set the title in bold face (and the current size) unindented
- don't indent the first line of the text after.

Furthermore, I would like to reuse the title in a running head or a ToC.

What are the problems?

The markup language constructs have to be chosen. In macro expansion the two-part macro T_EXnique should be the basis, IMHO, with all respect. In BLUe's format I adopted the pairs `\begin<tag>` and `\end<tag>`.

In order to set the title loose from the context we have to determine values for the glue, and be sure that it disappears at the top of a page. The coding solution below has different values for the parameters as supplied in `\beginsection` of The T_EXbook. From experiments it turned out that the values supplied by Knuth don't give nice results with BLUe. Too much glue was inserted for my taste.

Another problem is to prevent the head title to be printed at the bottom of a page on its own. The latter is related to the 'widow-orphan' phenomenon.

The coding of the minimal markup variant without parsing but with processing on-the-fly of 'the argument' has to be resolved in some elegant way.

Required greyboxes knowledge

In principle awareness is needed of

- two-part macrowriting
- MVL splitting details⁹

I consider it more pragmatic to start from a template by Knuth – `\beginsection` The T_EXbook 355 – where the design considerations have been taken into account. I made it a little more modular by starting from two-part macros

⁸. My philosophy on coping with change and variety is to separate stable issues from volatile aspects. I found that a User Interface is most susceptible to variety and change, especially when it is not as simple as simple can be, paradoxically as it may seem. For T_EX and friends this means that UI's vary and conversions have to be made, where the functionalities of the macroscopics commands vary too, making full-proof automated conversion impossible. Lowerlevel T_EXnical issues have been proven rather stable, and could be used in any flavour of T_EX.

⁹. This has all to do with how the OTR – Output Routine – splits the main vertical list, MVL for short, and is a key issue of T_EX's mapping copy onto pages.

and by introducing the token variables `\pre<tag>` and `\post<tag>`, supporting the dissection of concerns adago. They parameterize the placement of a document element within context. Whether you talk about titles, displays, tables, graphs, or you name it. Perhaps this should be generalized in something like `\pasteup<tag>`. Important!

Codes

For me two-part macros are at the heart, with one-part UI markup tags on top, one minimal. As can be seen below the two-part macros setup can be written without detailed knowledge in a functional spirit. Details are postponed.

```
\def\beginhead{\the\prehead\bgrouphheadfont}
\def\endhead{\egrouphthead\posthead}
%with auxiliaries
\prehead{\vskip0pt plus2ex
\penalty-250\vskip0pt plus1ex
\bigskip\noindent}
\posthead{\medskip\nobreak
\noindent\ignorewhitespace}
```

The `\prehead` replacement text has been borrowed from Knuth, and `\ignorewhitespace` has been communicated by Phil Taylor at a BachoT_EX some years ago.

The one-part markup form *with the same functionality* – nearly minimal because of the curly braces – reads

```
\def\head{\beginhead\bgrouph
\aftergroup\endhead
\afterassignment\ignorespaces
\let\dummy=}

```

Explanation. The replacement text starts `\beginhead` and a group. Immediately after the opening of the group `\aftergroup` takes care that the second part of the two-part macro will be invoked at the right place. The opening brace after the invocation of `\head` is read away by assigning it to `\dummy`. Spaces after the opening brace are ignored, due to `\afterassignment\ignorespaces`. The T_EXnique – moving the group opening – has been borrowed from plain’s `\footnote`, The T_EXbook 363. There it is used at the end of the replacement text of `\fo@t`.

The straight minimal markup macro reads as follows.

```
\def\bluehead#1\par
{\beginhead#1\unskip\endhead}
```

Note that here the processing on-the-fly functionality has been lost at the expense of minimal markup.

Example (Use of head macro)

In practice I use most of the time `\bluehead` with a blank line to separate the heading from what follows. But when

processing on-the-fly is wanted¹⁰ I use the more powerful lower level macros as shown below.¹¹

```
\beginhead|Head text
in verbatim|\endhead
Text after\\next line
```

```
\head{Head text, with |this|
in verbatim}
Text after\\next line
```

with results

```
Head text in verbatim
Text after
next line
Head text, with this in verbatim
Text after
next line
```

In general we have to store the title for reuse in a running head or a ToC, where the title is usually set in different fonts. When we store within T_EX we have lost the processing on-the-fly capability. To store material in a file and read it again when needed maintains the processing on-the-fly functionality. Here we have touched upon another issue which should be known about (La)T_EX use. How are ToCs done? When no file is used one can’t expect that material will be processed with suitable catcodes, that is with possibly different catcodes.

And the alternatives?

They suffer generally from

- too many ways to provide arguments, and why should we allow arguments at all
- general parsing technique of arguments.

What to do as macrowriter when creating your own push-the-button level? Study these macro collections – it will take you some time for sure – or simplify matters and stay with Knuth and let the `\(blue)head` be followed in the markup by just the title and a blank line and that is it, for most if not all of the cases.

4 T_EX inside in sight?

We have Knuth’s – much overlooked manmac by the way – macros, AMS (and TUG) macros, next to L^AT_EX, to name but three streams. These sets are highly incompatible. To

10. I personally don’t like much the use of a variety of fonts, especially not in the heading. I don’t like fontitus.

11. Within this L^AT_EX context `\beginlines` and `\endlines` have been simulated, next to the actual heading macros of BLUE.

use macros from AMS (TUG) is near to impossible because they are tied up with general approaches such as a common parsing philosophy, unless the macros which embody the basic philosophies are taken over as well, creating a too big an overhead not to mention the lack of understanding.

To adjust the OTR of L^AT_EX is not trivial to say the least. I would not trust the result in general because I for example can't tell whether it has influences on something elsewhere.

The disease of (non-Knuth) macros was, and probably still is, that once a macro (set) has been released it is immediately followed by a newer release to adjust for the bugs or insufficiencies.

Powerful T_EX features, akin to EP, have been overlooked at large. Let me mention only three items

- non-alien optional arguments functionality
- two-part macros
- active list separator.

Many a user got broke with the unusual `\expandafter`, `\futurelet`, `\afterassignment`, and `\aftergroup`. They are just coding auxiliaries to realize general functionalities, and of course a macrowriter should be aware of these, in short should know how to use the mentioned control sequences.

4.1 Optional arguments

This concept was known in high-level programming languages—HLPLs, for short—as well as in command languages, the pre-window one-liners to instruct computers. Maybe because of programmers being familiar with those, they have incorporated them and overlooked Knuth's new approaches. He has used token variables of which the text can be used—remember that T_EX is all about text replacement—can be inspected without parsing overhead, as simple as that. Next to Knuth's `\every...` I introduced `\this...` I used it on many places, for the first time in my `\btable`—bordered table—macros. There I abstracted into attributes such as `\framed`, `\(h/v)ruled`, and so on.¹²

Example (*Numbering lines in verbatim*)

How to implement the option of line numbering in verbatim? I did it by a definition `\numvrb`—called attribute in the SGML world, or simply flag—which can be set by its invoke in a `\thisverbatim`. Default `\nonum` is on. Next to this option we can think of other 'options', for example reading a file verbatim. And how to cope with semi-transparent verbatims?

```
\def\numvrb{\vrbmlin0
\everypar{\advance\vrbmlin1
\llap{\sevenrm\the\vrbmlin\quad}}}
```

```
%use
\thisverbatim{\numvrb}
\beginverbatim
...
!endverbatim
```

Note that BLUe's verbatim macros work on a line-by-line basis and that the line numbers can be adjusted via modifying the counter `\vrbmlin`. The `!` is the default escape character in BLUe.¹³

Knuth's handling of options

A beautiful example of how Knuth handled options is in manmac's `\begindisplay` and `\enddisplay`. Not via `\every...`—Knuth's main way of coping with options—but differently. It is used in The T_EXbook 29. All what follows after the opening tag `\begindisplay` up to the end-of-line is taken as '(optional) argument,' and inserted in the alignment display between `$$` and `\halign`.

No curly braces nor other opening and closings like `[` and `]`, or parsing. And when you don't need the option functionality you won't notice. It does not hinder nor impose on you. Minimal markup? Yes, definitely!

```
\begindisplay\hbadness10000
\hbox spread-.666667em{The badness
of this line is 100.}&
\quad(very tight)\cr
...
\enddisplay
```

This example shows what Knuth had in mind with respect to options and explains why he did not introduce `\this<tag>s`. The coding is a little more complicated but systematic.

Can it be applied throughout a format? With `\beginverbatim` I stumbled upon problems in applying options to the in-line verbatim with markup `[...]`. Needless to say that it went smooth via `\thisverbatim`.

4.2 Two-part macros

This concept was new for me until some 7 years ago. It does not occur in the HLPLs I'm familiar with. Knuth uses two-part macros but did hardly talk about them in The T_EXbook. In L^AT_EX's macros it is completely absent, at least in the L^AT_EX I looked into some 10 years ago. In using two-part macros your personal implementation can

12. At that time I had not adopted a general philosophy about handling of options, the `\this...` approach. I adopted the latter when I developed BLUe. So, some non-uniformities are in BLUe. The few attributes in `\btable` seemed so natural to me, that I did not change those, at the expense of non-uniformity.

13. The details of of BLUe's Verbatim have been explained in MAPS 94.1, available also on NTG's 4AllT_EX CD-ROM.

become simpler. In pursuit of Knuth I have used them abundantly in LATEX. This is a consequence of the catcodes idea, especially how and when catcodes are fixed. It comes down to when you parse an argument the catcodes are fixed while reading the argument. With two-part macros catcodes are fixed on-the-fly. This comes in handy when creating verbatims with an escape character, the so-called semi-transparent verbatims.

In Chapter 20 of The TEXbook there is no treatment on two-part macros, nor is there an entry for it in the index, alas. Exercise 5.7 deals with named blocks and checking of them. The latter is used in LATEX to make sure that the right environment closing tag is used in the markup. In Appendix E, where example formats (o.a. manmac) are explained, two-part macros are abundant. From the Appendix the following.

```
\beginchapter... \endchapter
\beginlines... \endlines
\begindisplay... \enddisplay
\beginnt... \endnt
\beginmathdemo... \endmathdemo
\beginchart... \endchart
\beginsyntax... \endsyntax
\begindoublecolumns... \enddoublecolumns
\exercise... \answer... \par
```

Furthermore, in the past questions were posed on TEX-nl, which exposed unfamiliarity with two-part macros. All this was enough for me to spend a paradigm column on two-part macros.¹⁴

Example (`\beginlines... \endlines`)

The functionality is that the script in-between is processed line-by-line and the result is preceded and followed by an `\hrule`.

```
\def\beginlines{\par\beginngroup\nobreak
\medskip\parindent0pt\hrule\kern1pt
\nobreak\obeylines\everypar{\strut}}
\def\endlines{\kern1pt\hrule\endngroup
\medbreak\noindent}
```

In The TEXbook script this is combined with in-line verbatim.¹⁵

Explanation. The replacement text of `\beginlines` is processed, followed by the formatting on-the-fly of the inserted material (starting on the next line after `\beginlines`) up to `\endlines`. The replacement text of the latter finishes it up.

Unwanted breaks are avoided. The `\hrule` is set in the first part and in the second part. Opening and closing of the group must be done at the right place. The in-between script is processed with `\obeylines` on.

Why?

The need for bothering about two-part macros is that the enclosed script elements are processed on-the-fly, meaning with the right catcodes.

One-part vs. two-parts

One-part macros are explained in Chapter 20 of The TEXbook. They are used as a shortcut for the replacement text parameterized by at most nine arguments.

A two-part macro is different. The first part sets up the ‘environment’ followed by script elements and ended by the second part, to finish up the environment.

LATEX emphasizes the environment concept – as a logical issue, with checking for the required endtag – in for example

```
\begin{abstract}... \end{abstract}
\begin{center}... \end{center}
\begin{itemize}... \end{itemize}
\begin{picture}... \end{picture}
\begin{quote}... \end{quote}
\begin{tabular}... \end{tabular}
\begin{thebibliography}...
\end{thebibliography}
%etc.
```

Example (*Catcodes*)

To digress a little on the above the following hypothetical example about the use of `\begindemo* \enddemo` versus `\demo*`. The purpose is to demonstrate that `*` will show up differently. The former is implemented as a two-part macro with processing on-the-fly capabilities, and the latter is implemented as a straight (one-part) macro with an argument. Suppose we have

```
{\catcode`*=13
\gdef\begindemo{\beginngroup
\catcode`*=13 \def*{MUL}}
\gdef\demo#1{\catcode`*=13 \def*{MUL}#1}
}\let\enddemo\endngroup
```

then the result of

```
\begindemo* \enddemo
%and
\demo*
```

14. See ‘Paradigm: Two-part macros,’ MAPS 95.1, available also on NTG’s 4 AllTEX CD-ROM. Note that Knuth’s `\beginchapter` does not process the title on-the-fly.

15. To set text verbatim. By the way, this is another approach to ‘verbatims with an escape character.’

differs. The first yields MUL and the latter *.¹⁶

Explanation. In the two-part case all is straightforward. While in the one-part case the * as argument is processed with the catcode at the moment of invocation, usually differently from 13, a token different from the one which is redefined in the replacement text.

One-part on top

Once we started from two-part macros we can build a one-part on top with the same functionality, albeit the tag must be followed by its argument(s) surrounded by curly braces. This can be achieved – also by T_EX – in the following meta way.

```
\def\tag{\begintag\bgroup
\aftergroup\endtag\let\dummy=}
```

Note that in the above example we could have complicated it more by introducing \demo{*}, with this \demo equivalent to the two-part macro.

eqalign as two-part macro

As an example of how to cast a one-part macro into two parts, and a one-part macro¹⁷ on top, let us rewrite \eqalign, The T_EXbook 362. The extra functionality of this approach is that the two-part variant can be used in those cases where the argument needs to be processed on-the-fly.

```
\def\beginqalign{\, \vcenter\bgroup
\the\thisqalign\openup1\jot\m@th
\starteqalign}
\def\starteqalign{\ialign\bgroup
\strut\hfil$\displaystyle{##}$&&
$\displaystyle{{}##}$\hfil\cr}
\def\endqalign{\cr}\egroup\egroup}
%with the one-part
\def\eqalign#1{\beginqalign
#1\endqalign}
```

I don't have a concrete example for the need for modifying \eqalign towards processing on-the-fly.¹⁸ However, it illustrates how to rewrite a one-part macro into two-parts as basis.

4.3 The active list separator

This functionality is used abundantly in BLUe, especially in selective loading of a tool, picture, reference, or address from a database. I used it for the first time in the Tower of Hanoi problem to be played and set by T_EX.¹⁹ By the way Knuth used the active list separator in The T_EXbook script, namely the \ansno to set the answers of the exercises after having written them first verbatim to a file.

Addresses, pictures and references

The entries of these databases obey the syntax

```
\lst<name>{<entryproper>}
```

that is the list consists of groups of two elements each preceded by the (active) separator.

Selective loading is achieved by properly defining \lst. Moreover, the names of the entries to be selected must be defined with whatever you wish as replacement text.²⁰

```
\def\loadselectivefrom#1{%
#1 = address lit pic
\def\lst##1{\ifx##1\undefined\ea\gobble
\else \ea\gdef\ea##1\fi}
\input #1.dat \relax%e.g. lit.dat
}
\def\gobble#1{}
```

Because of the scanning \outer defs are not allowed, nor are \par-s. The selective loading macro is embedded in the user macros \references and its ilks. In detail the meaning of 'loading' is adapted to the application. For references this means that the specified entries are set in a box and their names redefined by numbers. The names can be used for cross-referencing purposes while the box can be pasted up at your place of choice.²¹ However, the underlying searching methodology is the same for addresses, references and pictures.

Example (Loading and using a picture)

Loading a picture can be done simply as follows where I have abstracted from the filing system and therefore the markup is independent from the computer operating system.

```
\bluepictures \<name>pic, ...
```

The use is simply the invoke of \<name>pic, eventually preceded by \thispicture{...} to account for another scaling, the level of detail and so on.

16. A robustness aspect is to use \begingroup and \endgroup instead of \bgroup and \egroup, to facilitate the location of (user) unmatched braces in between.

17. Not more limited than the one available.

18. Although Knuth was not consequent, he apparently was right not to pursue things unduly.

19. For more details see 'Paradigms: Searching,' MAPS 96.2, available also on NTG's 4AllT_EX CD-ROM.

20. This approach is the opposite of preventing reloading. We tacitly want to redefine the fancy entries by the meaningful ones. My fancy text to be replaced is an error message.

21. Provided the box is not too large, otherwise it should be unboxed first.

How to create a picture for inclusion in the database ready for reuse is another story.²² For reuse detailed knowledge is no longer necessary. All that has to be remembered is the name of the picture, how to include it, and how to adjust it, that is how to override the default settings.

4.4 Different font sizes

The L^AT_EX world abstracted from the fonts via the use of the NFSS (New Font Selection Scheme) tool.

Again detailed knowledge is not necessary when one starts from Knuth's size-switching macro template, Plain T_EXies should copy and edit Knuth's size-switching macros – \tenpoint... – from The T_EXbook 414.²³

4.5 Only a few times I needed inside knowledge

There were a few occasions where I really needed advanced inside knowledge. One was for typesetting partitioned matrices with braces along its sides of *different length* but *equal thickness*. Jörg Knappen drew my attention to this aspect and provided a real hackery solution due to Alan Jeffrey.²⁴

On another occasion I was troubled by how to hyphenate accented words automatically and *completely*. The part after the accent is usually not hyphenated with the default accents. Bernd Raichle and Bas Romeny provided satisfactory explanations and ways around. The explanation is that an accent introduces *non-implicit* kerns before and after, and hyphenation stops at such kerns and therefore the rest is not hyphenated. But, one can with inside knowledge appropriately redefine the accent in such a way that T_EX3... will hyphenate the complete word.

5 Conclusions

I think for the easy use of (La)T_EX with T_EX toolboxes and macros knowledge of T_EX's inside should not be necessary. Insight in T_EX is mandatory, alas. Too much of this is the Achilles' heel for real wide-spread use of (La)T_EX.²⁵

Insight issues which have to be known are enumerated in the note.

Because of the above an easy, push-the-button, robust and reliable use of (La)T_EX and its accompanying tools is not yet in sight. Even worse, most likely I'm incomplete about what has to be known.

The add-ons don't compensate for this either, because they comprise generally mammoth collections, with the absence of small, concise and confined macros which perform one well-defined task to be combined into bigger tasks.²⁶

History has it that the latter has been recognized as useful in creating numerical program libraries, in UNIX' little languages and pipes, and in the RISC architecture of computers.

To make the circle round knowledge of T_EX's inside is mandatory for those who go for designing and creating the successor of T_EX, the ϵ -T_EX *casu quo* NTS²⁷ activities.

Because of the inherent complexity of the typographical tasks, a push-the-button successor of T_EX is not yet in sight.²⁸

Make your choice:

either master T_EX inside, or gain T_EX insight.²⁹

My case rests. Have fun, and all the best.

22. See the PWT guide, or the note Paradigms: The winds and halfwinds, MAPS 96.1, also available on NTG's 4AllT_EX CD-ROM.

23. It is said that scaling does not yield the best results. For optimal results we should run METAFONT again with adjusted parameters. T_EX's inside knowledge should be complemented with knowledge about METAFONT's inside – and some extras to find out and decide about what is optimal.

24. Example omitted in the L^AT_EX version of this note. See the Publishing with T_EX guide, or the NTG PR set (but there it is still wrongly displayed.)

25. Of course, we all take this for granted because of the lack of a comparable or better alternative.

26. I know, people would object and say that there is somewhere a L^AT_EX style file for so and so. But that is not general and restricted to L^AT_EX. I don't like L^AT_EX's design, nor do I wish to use it. I have created my own set which I understand, which is invariant to a high degree and suited for the purpose of the markup and typesetting of my notes for the rest of my life. T_EX is my EP lingua franca, comparable to English as a communication tool.

27. NTS stands for New Typesetting System initiated by DANTE. ϵ -T_EX – evolving T_EX – is the more modest and realistic spin-off approach towards the upward compatible successor(s) of T_EX. Both projects are directed by Phil Taylor. Peter Breitenlohner and Bernd Raichle are at the heart of ϵ -T_EX, to name but two knowledgeable wizards on the project. This makes me believe that it is a worthwhile effort, that a useful result will see the light of day some time. However, I can live with T_EX-as-is and the proposed changes I have heard off so far are not really necessary for me. It is a well-known social law that when an ϵ improvement is done it does not provide enough reason to follow the transition because of costs entailed by change, if not for the incompatibility disadvantages. But as a research/development project it is mandatory. For a status report see Phil Taylor's report for example in TUGboat, 18.1.

28. For me a T_EX script as a *useful* hypertext is not yet in sight either. Of course, the idea of a hypertext is nice, but I prefer, for the moment and near future, the selection and composition done by an author, instead of jumping through hyperspace and get some meaning out of it. However, jumping from a ToC or index entry towards the appropriate place is nice for a reader. As author I use ToCs differently. A ToC provides me a means to watch over structure when writing on-the-fly, as I usual do, starting from vague ideas of what I have on my mind. By writing them down and elaborating on them they become explicit and ready for development, communication and discussion.

29. At BachoT_EX'97 I learned another wordplay from Phil Taylor: 'incite,' for example 'insults incite resentments,' or more to the point 'feedback incites improvements.'

Bijlage 11

Local Guide T_EX-installatie Fewec

Maarten Gelderman
mgelderman@econ.vu.nl

abstract

Dit is de Local Guide van de L^AT_EX installatie van de Faculteit der Economische Wetenschappen en Econometrie van de Vrije Universiteit te Amsterdam. De Local Guide is niet bedoeld als L^AT_EX-handleiding, maar als aanvulling daarop. Er wordt een korte uitleg over L^AT_EX gegeven, een aantal bijzonderheden van onze L^AT_EX-installatie wordt genoemd en de gebruikte editor (μ ED) wordt besproken. De Local Guide kan zeker worden uitgebreid. Eenieder die nieuw materiaal heeft kan dit aan mij doorsturen en ik zal zorgen dat het, zoveel mogelijk ongewijzigd, wordt opgenomen.

1 Inleiding

Deze notitie is geen L^AT_EX-handleiding. Een korte L^AT_EX-handleiding is in Postscript-formaat beschikbaar onder `e:\tex\emtex\doc\short.ps`. In dezelfde directory vind je het bestand `graphics.ps`, met uitleg over het opnemen van (Postscript) afbeeldingen in L^AT_EX-documenten. Als je serieus met L^AT_EX aan het werk gaat kan je echter niet zonder de boeken van L^AT_EX [3] (basisboek) en Goossens, Mittelbach en Samarin [1] (voor 'gevorderden'). Verdere bruikbare informatie (bij voorbeeld de documentatie van alle gebruikte programmatuur) kan worden gevonden op CTAN (`ftp.dante.de`). De gebruikersondersteuning voor T_EX en L^AT_EX is goed. Er is een speciale Nederlandstalige ListServ lijst (stuur een mailtje met de tekst "subscribe voornaam achternaam" naar `listserv@nic.surfnet.nl`). De usenet-groep `comp.text.tex` is eveneens zeer actief en benaderbaar via bij voorbeeld NetScape (verdere uitleg hieromtrent kan je krijgen bij systeembeheer).

documentatie

Deze tekst beschrijft *wel* op welke fronten de FEWEC-installatie afwijkt van standaard T_EX-installaties: wat werkt niet, wat werkt anders en wat is er aan extra's beschikbaar. Tevens worden (ten dele) de lacunes ingevuld die ontstaan doordat in de bovengenoemde handboeken frequent wordt verwezen naar de Local Guides van de locale installatie. Sommige delen van de Guide zijn, zeker voor beginnende gebruikers, niet interessant. Alleen voor ervaren gebruikers van belang zijnde gedeelten van de tekst zijn in een klein lettertype gezet.

De eerstvolgende paragraaf van deze tekst bevat reclame voor T_EX. In paragraaf 3 wordt een algemene beschrijving van L^AT_EX gegeven. Het doel is nieuwe gebruikers kort aan te geven wat zij kunnen verwachten. Het gebruik van T_EX is essentieel anders dan het gebruik van een gewone tekstverwerker zoals bij voorbeeld WordPerfect. Een beginnend gebruiker zal met name moeite hebben met het doorzien van dit verschil en de nieuwe mogelijkheden die door deze opzet ontstaan. Het is duidelijk dat het niet zal lukken dit in één paragraaf aan te geven, ik zal echter toch een poging wagen.

opbouw tekst

Bij het installeren van software worden altijd keuzes gemaakt. De keuzes die gemaakt zijn bij onze installatie, en de consequenties hiervan, worden in paragraaf 4 aangegeven. Vervolgens worden de beschikbare commando's globaal besproken. In paragraaf 6 wordt relatief uitgebreid ingegaan op het gebruik van de standaard geïnstalleerde editor, daar hier elders niet direct informatie over beschikbaar is. Het is echter mogelijk om iedere willekeurige andere editor te gebruiken.

Dit is de letterlijk overgenomen tekst van de Local Guide van de faculteit van de Vrije Universiteit Amsterdam

In deze Local Guide wordt geen consequent onderscheid gemaakt tussen $\text{T}_{\text{E}}\text{X}$ en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Ik volsta met de opmerking dat $\text{T}_{\text{E}}\text{X}$ het eigenlijke programma is, terwijl $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ beschouwd kan worden als een AddIn of PlugIn voor $\text{T}_{\text{E}}\text{X}$ die extra functionaliteit ter beschikking stelt. Voor de beginnende gebruiker is dit onderscheid niet van belang. Wanneer je zelf extra functionaliteit wilt gaan ontwikkelen, zal je je ook in de werking van $\text{T}_{\text{E}}\text{X}$ moeten gaan verdiepen. Het boek van Knuth [2] is het voor de hand liggende startpunt.

2 Reclame

$\text{T}_{\text{E}}\text{X}$ is een ‘typesetting’ systeem, of in normaal Nederlands een geestelijk gestoorde tekstverwerker, dat in eerste instantie door Donald E. Knuth en een aantal andere medewerkers van Stanford is ontwikkeld. De belangrijkste voordelen van $\text{T}_{\text{E}}\text{X}$ zijn naar mijn mening:

1. De enige tekstverwerker ter wereld waarmee ‘alle mogelijke formules’ aan te maken zijn. $\text{T}_{\text{E}}\text{X}$ is het standaardplatform van de American Mathematical Society.
2. Het loopt nooit vast. Althans, het is mij nog nooit gelukt om het te laten crashen. Als het wel vast zou lopen, kan je contact opnemen met Knuth. Voor je dit doet is het echter verstandig om je wat verder te oriënteren. Zeker wanneer $\text{T}_{\text{E}}\text{X}$ opmerkingen in de trend van ‘I am stimifed’ (of groffer) gaat maken, is het verstandig de schuld bij jezelf te zoeken. In andere gevallen is het handig is de support-groepen (zie pagina 87) te raadplegen.
3. Het opnemen en maken van (technische) afbeeldingen gaat probleemloos. Zie het bestand `graphics.ps` voor verdere informatie.
4. Het eindresultaat ziet er prachtig uit. Dit komt met name door de geavanceerde afbreekroutine en het gebruik van ligaturen (bij voorbeeld ff, fl, fi in plaats van ff, fl en fi, ook æ en œ zijn natuurlijk erg æsthetisch op hun plaats gemaneuvreerd).
5. Literatuurverwijzingen en de literatuurlijst worden door $\text{T}_{\text{E}}\text{X}$ automatisch aangemaakt. Gevolg: minder typewerk (je hoeft een verwijzing maar één keer in te geven en kan hem vervolgens altijd gebruiken) en minder fouten.
6. Ook inhoudsopgaven, indices en dat soort lijsten worden automatisch aangemaakt. Het zelfde geldt voor verwijzingen naar pagina’s, figuren en paragrafen.
7. De lay-out van een document is altijd consequent en gemakkelijk als geheel te veranderen.
8. Er is een enorme hoeveelheid hulpmiddelen beschikbaar.
9. $\text{T}_{\text{E}}\text{X}$ draait op bijna ieder computerplatform. Gevolg: in geval van nood kan je altijd bij SARA terecht en (belangrijker) een andere $\text{T}_{\text{E}}\text{X}$ -gebruiker kan altijd met jouw teksten werken. Hierbij komt dat $\text{T}_{\text{E}}\text{X}$ in principe gratis is. Via Internet kan je versies downloaden. Ook zijn redelijk goedkope CD’s met $\text{T}_{\text{E}}\text{X}$ in de handel. De Nederlandse $\text{T}_{\text{E}}\text{X}$ Gebruikersgroep brengt onder andere een dubbel-CD uit met een kant-en-klare $\text{T}_{\text{E}}\text{X}$ installatie en een hele hoop hulpprogramma’s en documentatie erop.

Een aantal andere punten kan je zowel als een voor- als een nadeel zien:

1. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ werkt met zogenaamde markup. Dit is te vergelijken met het ‘onderwaterscherm’ van WP, maar dan tot in het extreme doorgevoerd. Je geeft niet aan dat iets vet en groot moet zijn, maar dat het een hoofdstuktitel is, etc. Dit dwingt je om logische teksten te schrijven. Voor het schrijven van een kort memo vinden veel mensen $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ echter te complex.

2. Een aantal uitgeverijen (o.a. Elsevier, Kluwer) gebruikt $\text{T}_{\text{E}}\text{X}$ voor het maken van boeken, artikelenbundels en tijdschriften. Door bovenaan een $\text{T}_{\text{E}}\text{X}$ -document (b.v.) aan te geven dat de documentclass `elsart` (elsevier artikel) is staat het automatisch in de standaard lay-out van Elsevier en kan de uitgever het direct verwerken. Dit is zowel een voor- als een nadeel, want er zijn natuurlijk vele uitgevers die geen $\text{T}_{\text{E}}\text{X}$ -documenten accepteren.
3. $\text{T}_{\text{E}}\text{X}$ werkt met ASCII-bestanden. Voordeel is de uitwisselbaarheid en robuustheid. Verder kan je (maar daar moet je een beetje voor freaken) een hoop bewerkingen direct op de file uitvoeren. Mogelijk nadeel: het is niet WYSIWYG.

3 Het werken met $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

In de vorige paragraaf is het meest wezenlijke verschil tussen $\text{T}_{\text{E}}\text{X}$ en andere tekstverwerkers tot het laatst bewaard: $\text{T}_{\text{E}}\text{X}$ werkt met ASCII-bestanden. Het bijzondere aan ASCII-bestanden is dat er niets bijzonders aan is: een ASCII-bestand is door vrijwel ieder willekeurig programma in te lezen en te gebruiken. Met bij voorbeeld het DOS-commando `type` kan een bestand direct op het beeldscherm worden weergegeven—met bestanden van een gewone tekstverwerker is dit niet mogelijk. Dit heeft wel tot gevolg dat er afspraken moeten worden gemaakt over de wijze waarop zaken die niet als gewone letters te beschouwen zijn—bij voorbeeld diacrische tekens (ö, õ, ô), symbolen (§, ¶, ©) en onderdelen van de layout (vet, cursief, lettergrootte)—in de tekst worden opgenomen.

De algemene afspraak bij het gebruik van $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ is dat alle functies worden opgeslagen als eenvoudige tekst, voorafgegaan door een backslash (`\`), en dat accolades (`{` en `}`) worden gebruikt om grenzen aan te geven (vergelijk haakjes in formules). Indien **vet woord** vet dient te zijn, kan in het bestand `tekst` met een `\textbf{vet woord}` erin worden aangetroffen. Je kan dit vergelijken met het ‘onderwaterscherm’ van WordPerfect. `Textbf` staat voor text boldface. Alhoewel het mogelijk is de tekst bij het bewerken vet in beeld te laten verschijnen, is het gebruikelijk (en ook in onze installatie is dit zo) het $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -bestand direct te bewerken.¹ Een belangrijke reden hiervoor is dat de meeste $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -gebruikers het commando `\textbf` bijna nooit zullen gebruiken. Het kan wel, maar er zijn veel handiger commando’s beschikbaar (zie de hierboven genoemde handleidingen). Het commando `\chapter` dient bij voorbeeld om een item in de inhoudsopgave op te nemen, de tekst vet en groot te maken en op een nieuwe pagina te laten beginnen, voorafgegaan door het hoofdstuknummer, en met na de kop twee lege regels.

Hiermee is impliciet een tweede kenmerk van het werken met $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ geïntroduceerd, waar ik even op in zal gaan, voordat ik terugkom op de voordelen en consequenties van het werken met ASCII-bestanden: het is gebruikelijk om in $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ te werken met logische en niet met visuele commando’s. Het achterliggende idee is, dat het belangrijker is (en meer mogelijkheden biedt) om te weten dat iets een hoofdstuk is, dan om te weten hoe dat iets er uitziet. WYSIWYG staat in de ogen van de $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -freak gelijk aan ‘what you see is all you get’. Indien dit nodig is, kan de gebruiker zelf een commando aanmaken om een structurelement aan te duiden.

Maar weer terug tot de kern van de zaak: de consequenties van het werken met ASCII-bestanden. Allereerst moet dit bestand worden aangemaakt, dit gebeurt in één of andere editor (bij voorbeeld JED, zie paragraaf 6). Vervolgens kan het ASCII-bestand worden ‘gebruikt’. Zoals reeds aangegeven zijn deze bestanden heel gemakkelijk benaderbaar. Iedereen die een klein beetje kan programmeren kan zonder al te veel moeite een programma schrijven dat een $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -bestand inleest en hier iets mee doet. De kracht van

1. Op het netwerk staat ook een programma, Scientific WorkPlace, dat de voordelen van een WYSIWYG-interface en gewoon $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ probeert te combineren. De mogelijkheden zijn echter beperkter dan die van $\text{T}_{\text{E}}\text{X}$ zelf. Documenten die met SWP gemaakt zijn, kunnen later wel relatief gemakkelijk als normaal $\text{T}_{\text{E}}\text{X}$ -document worden gebruikt. Verdere informatie hieromtrent kan je krijgen bij systeembeheer.

commando's

logisch vs visueel

(hulp)bestanden

\LaTeX ligt in het beschikbaar zijn van vele (soms heel simpele) programma's die nuttige dingen doen met het basisbestand. Wanneer je met \LaTeX werkt, werk je in feite met een aantal programma's die de \LaTeX bestanden inlezen en vervolgens het bestand zelf en/of een nieuw (hulp)bestand opslaan. Zo zijn er programma's die automatisch een literatuurlijst genereren, programma's die een index aanmaken, programma's die uit een bestand met vele vragen automatisch een tentamen kunnen genereren, etc. En natuurlijk, het meest belangrijk van allemaal, zijn er de \TeX -compiler, de previewer en de printerdriver. De \TeX -compiler laadt eerst het hulpprogramma \LaTeX , en begint vervolgens met het verwerken van het \LaTeX -bestand dat de tekst met commando's bevat en de eventuele hulpbestanden—literatuurlijst, inhoudsopgave, index—die door andere programma's zijn gegenereerd. Vervolgens genereert de \TeX -compiler een bestand dat door de previewer kan worden gebruikt om het document op beeld te zetten en door de printerdriver kan worden gebruikt om het document te printen. Ook maakt de \TeX -compiler een aantal eigen hulpbestanden aan. Deze hulpbestanden kunnen weer door hulpprogramma's en de compiler zelf worden gebruikt. Hiermee is de cirkel rond. Met een concreet voorbeeld zal ik nog proberen het geheel iets minder abstract te maken.

Dit document is aangemaakt in JED. De eerste keer dat de \TeX -compiler het bestand inlas heeft deze niet alleen een bestand voor de previewer en de printerdriver aangemaakt, maar ook een aantal hulpbestanden. Eén zo'n hulpbestand is door het programma $\text{BIB}\TeX$ gebruikt om de literatuurverwijzingen uit mijn database met titels te halen. Een ander hulpbestand wordt door de \TeX -compiler zelf weer gebruikt om een volgende keer de inhoudsopgave en de verwijzingen (van de vorm zie paragraaf . . .) van het document aan te maken.

4 Opzet van de installatie

De \TeX -installatie op onze faculteit is opgezet met het doel voor iedereen een werkbare \LaTeX -omgeving te bieden. Vanzelfsprekend is deze installatie niet volledig, maar wanneer je tegen het ontbreken van bepaalde onderdelen aanloopt kunnen deze in principe op het netwerk worden geïnstalleerd. Hierbij dient het volgende voorbehoud te worden gemaakt. De installatie is zo opgezet dat deze door iedereen vrij kan worden gebruikt en gekopieerd. Er wordt alleen gebruik gemaakt van freeware. Ter voorkoming van problemen met copyrights wordt aan deze benadering vast gehouden: shareware en commerciële software worden niet aan de installatie toegevoegd. Verder worden de $\LaTeX 2_{\epsilon}$ -afspraken zoveel mogelijk gevolgd: verouderde en niet compatible modules worden *niet* aan de installatie toegevoegd. Zolang de ec-fonts niet in definitieve versie beschikbaar zijn, worden deze in principe niet op het netwerk geïnstalleerd (hier valt op basis van goede argumenten, b.v. de acute behoefte om Tsjechische karakters die niet in de standaard computer modern fonts voorkomen te gebruiken, over te praten).

Packages en style-files die voor eigen gebruik bestemd zijn, kunnen in de directory `c:\mytex` geplaatst worden, en zijn dan naadloos in de installatie geïntegreerd. Gebruikers die behoefte hebben aan het gebruik van een andere structuur kunnen het bestand `e:\tex\emtex\bin\texstart.bat` naar een plaats *voorin* het zoekpad kopiëren en aan hun wensen aanpassen.

Gebruikers die met plain \TeX en METAFONT wensen te werken zullen niet alles van hun gading vinden. De installatie is echter zo opgezet dat zij deze gemakkelijk lokaal (op hun eigen harde schijf of netwerkdirectory) uit kunnen breiden: kopieer de batchbestanden uit de directory `e:/tex/emtex/bin` naar een plaats *voorin* je zoekpad en pas ze aan je eigen behoeften aan.

Nog beter is het natuurlijk om zelf eigen batchfiles te maken die de batchfiles op het netwerk aanroepen en vervolgens enige eigen instellingen toevoegen. De opties zijn schier eindeloos, maar je bent natuurlijk wel aan je eigen kunnen overgeleverd. Gelukkig zal eenieder die niet meer met de netwerkinstallatie uit de voeten kan ook zelf wel in staat zijn om deze wijzigingen te maken of zijn eigen locale installatie op te zetten.

5 Beschikbare commando's

T_EX komt beschikbaar na het intypen van het commando `texstart`. Dit commando zorgt er voor dat een aantal instellingen van de gebruikte computer zo wordt aangepast dat **T_EX** gebruikt kan worden, tevens wordt een informatiescherm getoond. Voor het overige is er niets zichtbaar veranderd. Het commando dient alleen om de instellingen van de computer aan te passen, zodat de hieronder besproken commando's naar behoren functioneren. Met behulp van het commando `texstop` worden de oude instellingen weer teruggezet. De computer gewoon uitzetten is ook mogelijk: de instellingen worden nergens opgeslagen.

Alle commando's hebben de vorm `commando parameters bestandsnaam`. De bestandsnaam dient normaal gesproken *zonder* extensie te worden ingegeven. Om pagina 3 tot en met 10 van het document `piet` te printen type je `pps -p3 -l10 piet`. Indien er extra parameters beschikbaar zijn, kunnen deze meestal opgeroepen worden door de parameter `-?` te gebruiken. De belangrijkste uitzondering op de regel dat de bestandsnaam zonder extensie dient te worden ingegeven is het aanroepen van de editor. Daar de editor geen vast onderdeel van de installatie vormt dient de bestandsnaam met de extensie `.tex` te worden ingegeven.²

De volgende opties zijn op het netwerk beschikbaar (deze lijst is niet volledig, bekijk zelf de inhoud van `e:\tex\emtex\bin` om vast te stellen welke programma's er verder beschikbaar zijn). Indien een commando eindigt op **32**, is dit een 32-bits programma. Zulke programma's werken niet altijd op oudere computers, vervang **32** door **16** en alles hoort weer te draaien.

jed Bewerk het document, zie verder paragraaf 6.

latex Compile het document met **L^AT_EX**.

v Preview van het document op het scherm. De belangrijkste parameters van dit commando zijn `-b` eerste paginanummer en `-l` laatste paginanummer.

pps Print het document op een postscriptprinter. De postscriptprinter (te vinden in het printerhok en op het secretariaat econometrie) biedt de meeste grafische mogelijkheden. De belangrijkste parameters van dit commando zijn `-p` eerste paginanummer en `-l` laatste paginanummer.

php3 Print het document op een Laserjet III. De belangrijkste parameters van dit commando zijn `-b` eerste paginanummer en `-l` laatste paginanummer.

php4 Print het document op een Laserjet IV. De belangrijkste parameters van dit commando zijn `-b` eerste paginanummer en `-l` laatste paginanummer.

lacheck Controleer het document op syntaxfouten (het gebruik van bij voorbeeld verkeerd gespelde commando's) en stijlfouten (bij voorbeeld een spatie voor het vraagteken aan het einde van een zin).

bibtex32 Genereer literatuurverwijzingen

bibdb Houd database met literatuur bij

mkidx32 Genereer index

mytex Versnel het compileren van grote documenten ten koste van het gebruik van schijfruimte. Experimenteel, maak eerst een backup!!

2. Op deze manier is **JED** ook te gebruiken om bij voorbeeld **HTML**-bestanden te bewerken.

6 Bestanden bewerken: jed

editors

Zoals hierboven reeds is aangegeven werkt \LaTeX met ASCII-bestanden. Je kan voor het bewerken van \LaTeX -bestanden dus iedere willekeurige ASCII-editor gebruiken.³ In principe kan je zelfs gebruik maken van Wordperfect of een andere tekstverwerker en de tekst als ASCII-bestand opslaan. ASCII-bestanden worden in tekstverwerkers veelal aangeduid met de term DOS-bestand. Op het netwerk van de VU is JED geïnstalleerd. Deze editor is een kleine versie van emacs gecombineerd met \AUCTEX . Deze laatste combinatie staat bekend als de ideale omgeving voor het editen van \LaTeX -bestanden, maar werkt niet onder DOS.⁴ Eenieder die de mogelijkheid heeft om op emacs over te stappen raad ik aan dit te doen.⁵ Gelukkig verschaft JED een groot deel van de door emacs geboden functionaliteit. Evenals met emacs het geval is, is het ook met JED mogelijk om 'oneindig' grote bestanden te bewerken (met de combinatie $\text{\LaTeX}/\text{JED}$ is het geen probleem om je gehele proefschrift in één enkel document op te nemen) en de meest gebruikte functies zijn in JED onder dezelfde toetsen en namen te vinden als in emacs. Later alsnog overstappen is dus gemakkelijk.

Een belangrijk voordeel van emacs (en derhalve van JED) zijn de grote flexibiliteit van het programma en de efficiëntie waarmee tekst kan worden ingegeven. Deze voordelen hebben echter de bekende prijs: JED is een complexe editor en het kost behoorlijk wat inspanning om er daadwerkelijk efficiënt mee te leren werken. Ook is de interface in eerste instantie niet echt intuïtief. Na verloop van tijd ben je je er echter niet meer van bewust welke toetsencombinaties je gebruikt.⁶

toetsencombinaties

De in te geven toetsencombinaties zijn in de volgende paragrafen telkens in een vierkant weergegeven. Een afbreekstreepje geeft hierbij aan dat twee toetsen tegelijkertijd ingedrukt dienen te worden. Een hoofdletter A aan het begin van de toetsencombinatie staat voor de Alt-toets,⁷ een hoofdletter C voor de Ctrl-toets. Een hoofdletter F, gevolgd door een nummer van 1 tot en met 12 staat voor één van de functietoetsen boven aan het toetsenbord. In deze notatie betekent het (niet bestaande) commando $\boxed{\text{A-F12 C-q}}$ het volgende: toets eerste gelijktijdig de Alt-toets en de toets met F12 in en toets vervolgens de Control en de q toets gelijktijdig in.

6.1 Help functionaliteit JED beschikt over een aantal mogelijkheden om de (beginnende) gebruiker het werk te vergemakkelijken. De meest gebruikte functies zijn door middel van een archaïsche menustructuur te benaderen. De menu's worden opgeroepen met behulp van $\boxed{\text{C-h ?}}$. Let hierbij op de onderste regel van het scherm. JED probeert hieraan te geven hoe een commando direct kan worden aangeroepen. Blijven werken met de menu's is de manier om een bloedhekel aan JED te krijgen.

Een volgende help-functie is de info-functie. Het ingeven van $\boxed{\text{C-h I}}$ zou tot een rondleiding door de volledige functionaliteit dienen te leiden. De eerste mededeling die de gebruiker echter tegenkomt, is dat hij of zij vriendelijk verzocht wordt te helpen bij het completeren van de documentatie.

Het trio $\boxed{\text{C-h a}}$, $\boxed{\text{C-h f}}$ en $\boxed{\text{C-h w}}$ zal ook door de meer ervaren gebruiker gewaardeerd worden. Met behulp van het eerste commando kan een lijst worden opgeroepen van alle interne JED functies waarin een ingegeven woord voorkomt. Met behulp van het tweede commando kan een uitgebreidere beschrijving van één enkel commando worden opgeroepen en met behulp van

3. Omgekeerd kan JED gebruikt worden voor het bewerken van ieder willekeurig ASCII-bestand. Omdat HTML-bestanden ook gewoon ASCII-tekst bevatten is JED ook zeer geschikt voor het bewerken van bijvoorbeeld World Wide Web pagina's.

4. Emacs en \AUCTEX zijn wel onder DOS beschikbaar, maar niet alle functies werken; per saldo biedt JED meer functionaliteit en is sneller.

5. Voor OS/2, Linux, Windows95 en WindowsNT zijn versies van emacs te vinden op Internet.

6. Dit ontdekte ik eigenlijk pas toen ik deze notitie schreef en het merendeel van de toetsencombinaties niet op bleek te kunnen schrijven zonder ze eerst een keer langzaam in te geven.

7. In plaats van de Alt-toets mag ook $\boxed{\text{Esc}}$ gevolgd door de genoemde letter worden ingegeven.

het laatste commando kan worden nagegaan welke toetscombinatie gebruikt kan worden om het commando op te roepen.⁸

6.2 Buffers en bestanden Voor diegenen die geen complexe bewerkingen met JED uitvoeren volstaat het te weten dat JED een opgevraagd (tekst)bestand buffer noemt. Tegelijkertijd kunnen meerdere buffers geopend zijn. Op deze manier kan met meerdere bestanden tegelijk worden gewerkt. Alhoewel dit over het algemeen niet aan te bevelen is, is het ook mogelijk een bestand meerdere keren op te vragen, JED geeft geen waarschuwing als je dit doet, wel verschijnt er bij het tweede geopende bestand onder in beeld, achter de bestandsnaam een 2.

[C-x C-f] Open een bestand

[C-x C-s] Sla een bestand onmiddellijk op, vraag niet op bevestiging

[C-x s] Sla een bestand op, maar vraag eerst om bevestiging

[C-x C-w] Sla een bestand op onder een nieuwe naam

[C-x b] Ga na een andere buffer

[C-x C-b] Geef een overzicht van de geopende buffers

[C-x k] Sluit de huidige buffer

[C-x c] Beëindig het werken met JED

Zoals de lezer, waarschijnlijk heeft opgemerkt beginnen alle hier besproken commando's met **[C-x]**. Het tweede gedeelte van het commando, kan gemakkelijker onthouden worden door aan de betekenis van de letter (file, save, write, buffer, kill) te denken.

[C-x 2] Maak twee windows

[C-x 0] Ga (indien meerdere windows aanwezig zijn) naar het andere window

[C-x 1] Maak weer één window

[Esc C-c C-v] PageUp/PageDown in het niet actieve venster

[Esc !] Verlaat JED tijdelijk en ga naar een DOS-shell

6.3 De minibuffer De regel onder in beeld (onder de regel waarop tijd en datum staan), heet minibuffer en wordt gebruikt voor het weergeven van informatie en voor het vragen van informatie aan de gebruiker. Wanneer je bij voorbeeld een functie middels de (omslachtige) menu's van JED aanroept, verschijnt onder in beeld de (efficiëntere) toetsencombinatie waarmee het commando direct kan worden aangeroepen. Wanneer je commando's gebruikt waarvoor JED verdere invoer nodig heeft (bij voorbeeld zoeken en vervangen of het aanmaken van een environment) verschijnt hier een prompt voor verdere informatie. De meeste functies (waaronder de cancel-functie **[C-g]**) werken gewoon in de minibuffer.

In de minibuffer is ook een aantal bijzondere commando's beschikbaar. Met behulp van de pijltjestoetsen **[↑]** en **[↓]** kunnen eerdere minibuffers worden teruggeroepen. Dit is bij voorbeeld handig bij het oproepen van bestanden en bij de zoek en vervangfunctie. Met behulp van de Tab-toets kan een lijst worden opgeroepen van beschikbare mogelijkheden. Bij het openen van een bestand bestaat de lijst uit de beschikbare bestanden en directories. Met de spatiebalk kan door de lijst van mogelijkheden worden heengelopen. Door het ingeven van letters wordt het aantal mogelijkheden beperkt. Is in de minibuffer bij

8. Het omgekeerde is ook mogelijk, gebruik **[C-h k]**.

voorbeeld een 'a' ingegeven, dan worden alleen nog de bestanden weergegeven waarvan de naam met een 'a' begint.

Met `[Esc x]` wordt een bijzondere minibuffer aangeroepen. In deze buffer kunnen direct JED-commando's worden ingegeven. Op deze wijze is de volledige functionaliteit van JED beschikbaar. De Tab-toets en de spatiebalk werken op de hiervoor aangegeven wijze. Met behulp van `[Esc x Tab]` kan derhalve een lijst met alle interne functies van JED worden opgeroepen.⁹

6.4 Het bewerken van tekst Het bewerken van tekst in JED verschilt niet veel van het werken in andere editors. Met behulp van `[Insert]` kan gewisseld worden tussen insert-en overwrite-mode. De pijltjes toetsen dienen om door de tekst heen te lopen. Met behulp van `[Home]` wordt naar het begin, en met behulp van `[End]` naar het einde van de regel gesprongen. Ook `[PageUp]` en `[PageDown]` werken op de bekende wijze. Verder zijn de volgende commando's beschikbaar:

`[C-←]` Ga een woord naar links

`[C-→]` Ga een woord naar rechts

`[C-Home]` Ga naar het begin van het document

`[C-t]` Verwissel de letters links en rechts van de cursor

`[C-s]` Zoek vooruit vanaf de cursor

`[C-r]` Zoek achteruit vanaf de cursor

`[A-%]` Zoek en vervang

`[A-q]` Format paragraaf zorgt door inspringen en het afbreken van regels voor een leesbare tekst (in ASCII tekst komen geen zachte returns voor, hierdoor zou zonder dit commando de tekst al snel onoverzichtelijk worden). Dit commando is binnen zekere grenzen intelligent, maar werkt niet goed bij tabellen.

`[C-x u]` Maak de vorige actie ongedaan.

`[Esc nummer]` Prefix: herhaal, indien van toepassing, het volgende commando *nummer* keer.¹⁰

De volgende commando's hebben allemaal betrekking op het knippen, plakken en kopiëren van stukken tekst.

`[F12]` Maak een blok

`[Shift-F12]` Zet de huidige sectie in een blok

`[C-F12]` Zet de huidige environment in een blok

`[C-w]` Verwijder blok

`[C-k]` Verwijder tot einde van de huidige regel

`[A-w]` Kopieer blok

`[C-y]` Plak een eerder verwijderd of gekopieerd stuk tekst

9. In deze lijst verschijnen alleen de interne functies. Er zijn ook externe functies beschikbaar, maar deze worden niet weergegeven. Een voorbeeld zijn de speciale L^AT_EX-commando's die met behulp van een soort AddIn aan JED zijn toegevoegd.

10. Bij een aantal commando's heeft herhalen geen zin (bij voorbeeld bij het format-commando, het heeft geen zin de tekst tien keer goed te zetten). Bij deze commando's wordt de werking van het commando veranderd door de prefix. In het genoemde geval wordt de tekst tevens uitgevuld. `[Esc nummer]` gevolgd door `[C-x f]` stelt de uitvulkolom in.

C-c ; Zet het geselecteerde blok tekst om in commentaar

C-c : Zet het geselecteerde blok met commentaar om in gewone tekst (het omgekeerde van het vorige commando)

De volgende commando's veranderen het huidige woord, vanaf de plaats waar de cursor zich bevindt:

A-l Maak woord lowercase (alle letters vanaf de cursor worden kleine letters)

A-c Maak woord gekapitaliseerd (alleen de eerste letter van het woord wordt een hoofdletter)

A-u Maak woord uppercase (alle letters vanaf de cursor worden hoofdletters)

6.5 L^AT_EX-commando's Wanneer je een document opvraagt met de extensie .TEX schakelt JED automatisch over naar L^AT_EX-mode. Achter de bestandsnaam verschijnt dat het woord LaTeX op de informatieregel. Indien de naam van het te bewerken document niet de extensie .TEX heeft kan toch naar L^AT_EX-mode worden overgeschakeld, door na het ingeven van **Esc x** het commando latex-mode in te geven. In deze mode is een aantal commando's beschikbaar dat gebruikt kan worden om met een paar toetsaanslagen hele (series) L^AT_EX-commando's in te geven.

Lettertype veranderen

C-c C-f C-b **Vette** tekst

C-c C-f C-e *Benadrukte* tekst

C-c C-f C-i *Italic* tekst

C-c C-f C-u Rechtopstaande tekst

C-c C-f C-s *Schuine* tekst

C-c C-f C-f Sans serif tekst

C-c C-f C-c KLEINE KAPITALEN tekst

C-c C-f C-t Typemachine tekst

De lettertype-commando's voegen de voor het veranderen van het lettertype noodzakelijke L^AT_EX-commando's in in de lopende tekst. Om de één of andere duistere reden werken deze commando's in JED niet altijd goed—caveat lector—gelukkig treedt dit probleem in lopende tekst 'nooit' op.¹¹

Environments en indelingen

C-c C-e Voeg een environment in

C-c] Sluit environment af

C-c C-j Voeg een item toe aan een lijst

C-c C-s Voeg een indeling in

Overige commando's

11. De commando's bezitten een zekere intelligentie en proberen de gebruiker werk te besparen. Helaas heb ik deze intelligentie nog niet doorgond en komt het geheel daardoor niet helemaal uit de verf.

A-Tab loop (na het intypen van een begin) door het herhaald intoetsen van **A-Tab** door de \LaTeX -commando's die met de reeds ingegeven letters beginnen heen.

C-g Cancel

' Voeg een enkel aanhalingteken in; JED zorgt er voor dat links een “” en rechts een ”” wordt ingevoerd

" Voeg een dubbel aanhalingteken in; JED zorgt er voor dat links een “” en rechts een ”” wordt ingevoerd

\ Letterlijk overnemen van de volgende toetsaanslag. Op deze manier kun je toetsen waaraan een bijzondere functie is toegekend, toch blijven gebruiken om bij voorbeeld in plaats van “” de op de toets vermelde “n” in te voeren.

C-u Verander de werking van de font en de environmentcommando's zo dat de huidige omgeving wordt veranderd. Sta je bij voorbeeld midden in een itemize environment (lijst met bullets) en wil je deze veranderen in een enumerate environment (lijst met nummers), dan kan dit door in te geven **C-u C-c C-e**. Het commando voor het aanmaken van een environment (**C-c C-e**) wordt voorafgegaan door **C-u**.

7 Miscellaneous

Er is nog veel ander \TeX -gerelateerd materiaal geïnstalleerd op de E:-drive en te vinden op het Internet. Ik zal dat hier niet allemaal bespreken. Het belangrijkste zijn de Postscript-utilities en de conversion utilities. De Postscript-utilities kunnen gebruikt worden om willekeurige Postscript-bestanden (dus niet alleen bestanden die met \LaTeX gemaakt zijn, maar ook working papers van derden en Windows-print-bestanden) te bewerken. Zo is het ondermeer mogelijk om meerdere pagina's op één vel papier af te drukken met het programma `psnup`. Een aantal utilities met documentatie staat in de directory `e:\tex\emtex\psutils`.

Converteren van Wordperfect naar \LaTeX kan met behulp van het commando `wp2latex`. Dit programma neemt geen afbeeldingen e.d. mee bij de conversie. Geavanceerdere (niet-gratis) programma's voor conversie zijn gekoppeld aan Scientific WorkPlace.

Converteren vanuit \LaTeX kan onder andere met behulp van het programma `tex2rtf`. Hiermee kunnen RTF-bestanden (geschikt voor de meeste Windows-tekstverwerkers) en HTML-bestanden worden gemaakt van \LaTeX -bestanden. Bestanden dienen wel aan zekere eisen te voldoen om deze conversie vlekkeloos te laten verlopen. Installeren van `tex2rtf` doe je onder Windows met behulp van het programma `e:\tex\tex2rtf\install.exe`. Een conversie-programma dat wel altijd werkt is `dvitype`, maar de mogelijkheden hiervan zijn beperkter (\approx nihil).

Referenties

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The \LaTeX Companion*. Addison-Wesley, Reading, MA, USA, 1994.
- [2] Donald E. Knuth. *The \TeX book*. Addison-Wesley, Reading, MA, USA, 1984.
- [3] Leslie Lamport. *\LaTeX : A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, second edition, 1994.

Bijlage 12

User manual for `kluwer.cls`

Instructions for authors

Kluwer Academic
Publishers,
prepress Department
Achterom 119,
3311 KB Dordrecht,
The Netherlands
texhelp@wkap.nl

abstract

This is the user manual for the `kluwer.cls` class file for the preparation of articles to be submitted to journals or spin-offs of journals published by Kluwer Academic Publishers. The `kluwer.cls` is a generic class, which allows us to have a simple conversion to different journal layouts. The `kluwer.cls` class file is similar to the `article` class file of $\text{\LaTeX} 2_{\epsilon}$, but it has some extra fields in the preamble and some extended commands for the body text.

keywords

$\text{\LaTeX} 2_{\epsilon}$, electronic submission, user manual, class file, Kluwer

1 Introduction

The `kluwer` class file is the general style file that should be used when submitting an article in $\text{\LaTeX} 2_{\epsilon}$ for publication in one of the journals of Wolters Kluwer Academic Publishers.

Please note that the layout generated by the `kluwer` class file will deviate from the style of the journal in which your article will be published. The article will be formatted in the appropriate journal style by Kluwer after acceptance. In that stage, the copyright information will be added, the running heads will be adapted if necessary and the article will be typeset in the Times New Roman font.

2 Getting started

The overall structure of an input file is given in figure 1.

2.1 The Beginnings The `kluwer` class file is addressed by using the command: `\documentclass{kluwer}`. The file `kluwer.cls` and its subfiles should be accessible to $\text{\LaTeX} 2_{\epsilon}$.

The class file defines some options that can be inserted between angle brackets. These options will be introduced as we go along.

The date for this version of the class file is **1997/05/30**.

Amongst other things, the class file has already loaded the packages `amssym` and `wasysym` on startup. Support for ‘Here’ style floats, rotated floats and endnotes is built in.

2.2 The preamble In general, you should use standard \LaTeX commands as much as possible. If you create your own macros *always* put them in the preamble, and use \LaTeX syntax instead of `\def` whenever possible. Take extra care to make sure that we receive and/or can obtain *all* included package files used by your submission.

As a rule of thumb, we can find only what’s in a relatively recent “standard \LaTeX distribution”; anything else should be sent to us, together with your submission.¹

¹ This also includes outdated versions of currently available macro packages. For instance, if you use an old version of the `amstex` package, you should send those files as well, since we usually only have the latest versions of any package.

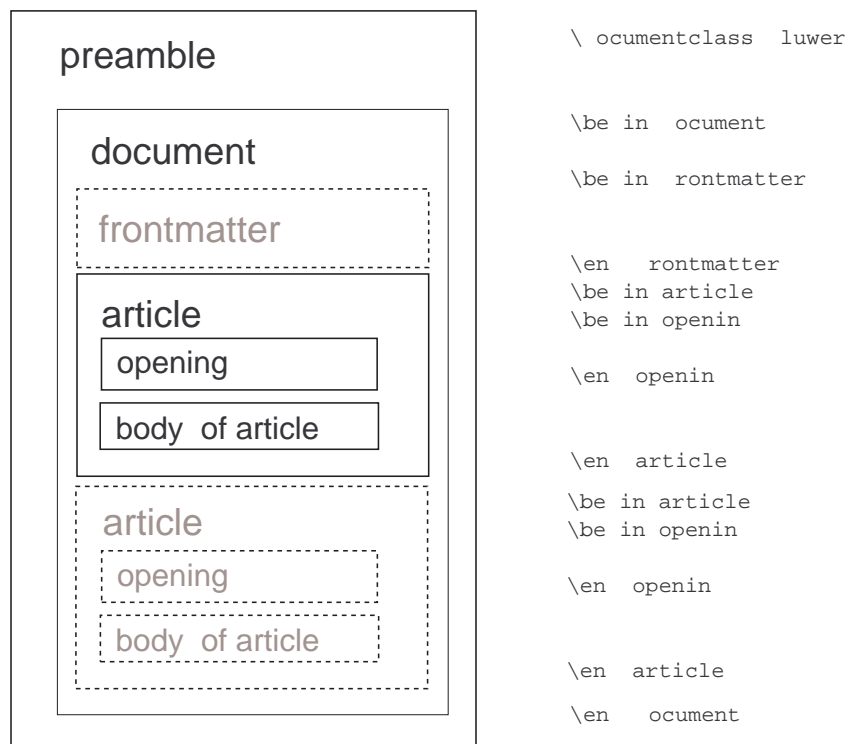


Figure 1. Overall structure for input files. Optional parts are displayed in grey (with dotted lines) on the left side.

3 The 'opening' environment

The title, subtitle, author names & institutes and the submission dates for the article are handled by the `opening` environment.

In Figure 2 you will find a list of the chain of commands that should be used only within the `opening` environment. Most of these commands will produce an error if used somewhere else in your document. The subsequent paragraphs will deal with the commands in detail.

The order of commands you find in the figure is not fixed, all commands are executed by `\end{opening}`, without regard to their location within the `opening` environment (with an exception for the author block, please see below for details). Anything that is not within a command will end up being typeset *before* the 'real' contents of the environment, so be careful to only use the commands given here.

Now let us explain the commands in more detail.

3.1 Title and Subtitle The title and, if necessary, subtitle should be typed in one of the following two styles:

```

\title{This is my title}      One initial capital
\title{This Is My Title}    Initial capitals

```

```

\begin{document}
\begin{article}
\begin{opening}
% Title block:
\title{...}
\subtitle{...}
% Author block [repeatable]:
\author{...\surname{...}\email{...}}
\author{...\surname{...}\email{...}}
\institute{...}
% Extra information:
\dedication{...}
\translation{...}
\runningtitle{...}
\runningauthor{...}
\begin{ao}
\end{ao}
\begin{motto}
\end{motto}
% Abstracts & keywords:
\begin{abstract}
\end{abstract}
\keywords{...}
\abbreviations{\abbrev{...}{...};
               \abbrev{...}{...}}
\nomenclature{\nomen{...}{...}; \nomen{...}{...}}
\classification{...}{...}
\end{opening}

```

Figure 2. Commands to be used within the opening environment

This has to be done according to the style of the journal in which your article is to be published. Uppercasing will be done by the stylefile if necessary. A missing `\title` command generates an error message.

You can split the typeset title and subtitle over multiple lines by putting two backslashes at an appropriate place, and you can specify footnotes to the title by use of the `\thanks` command. The specified text will be placed at the bottom of the page as a footnote.

An example of a title block could look like this:

```

\title{The 'KLUWER' LaTeX Style File:\\
       Instructions for Authors\thanks{Donald Knuth}}
\subtitle{Basic Instructions}

```

3.2 Author block The author block consists of one or more authors followed by an institute. This construction may be used more than one time. A typical 'large' author block would look like this:

```

\author{A. \surname{Thor1}\email{thor@wkap.nl}}
\author{B. \surname{Thor2}\thanks{Partially support...}}
\institute{KAP}
\author{C. \surname{Thor3}}
\institute{}

```

As you can see, the surname of the author is given within a separate command, `\surname`. The point of this is that this makes it easier to extract information from the \LaTeX file. It has no influence on the way the author's name is typeset.

You can specify support acknowledgements etc. by use of the `\thanks` command.

If the author has an email address that looks like it will last for at least the time between submission and publication, you can give the address in the `\email` command either after the surname of the author or in the `\institute` command. It will be typeset in small typewriter font within brackets.

The specification for the author block is a bit more complicated than the rest of the opening environment. Please take note of the following restrictions:

1. Order is important for all `\author` & `\institute` commands. They will be typeset in the same ordering as found in your input file.
2. `\thanks` can only be used as the very last item in the `\author` command. If you make a mistake, you will get a quite incomprehensible \TeX error message.
3. The list of authors *has* to end with an institute. If there are authors that are not part of any institute, give an empty command as demonstrated above. If you don't, that author's name will *not* be typeset.
4. Sometimes the class file may fail to find a good place to break a line containing multiple authors. If this happens, you can insert two backslashes yourself to force a line break, but this will *not* work if it is the very first command within `\author`. In this case there will be a \LaTeX error 'No line to end'.

3.3 Extra information These commands give some extra information about the article.

Dedicated and translated articles You can use the commands `\dedication` and `\translation` to give extra information about your article. The commands help by inserting the correct amount of space above and below your text.

Running heads You can create a running title (used on odd pages) by using the `\runningtitle{}` command, where you put the (maybe shortened) title between the curly braces, and you can create a running author line (used on even pages) using the analogous `\runningauthor` command.

Both of these commands may be expanded in strange ways by \LaTeX , so try to keep the information short enough to fit in the headline, and try to use only plain text. Using `math` (or any kind of command) in a running title will not always work. Especially, do *not* force line breaks.

Address for correspondence The environment `ao` can be used to give an address for correspondence or offprints. The contents of this environment will be typeset at the end of the article environment (an example is given in this document on page 111). The exact phrasing of the leading text depends on the stylefile, so it may change later here at Kluwer.

3.4 Abstracts and keywords The abstract environment is used to give a short description of the article.

The `\keywords` command gives a list of keywords. These should be separated with commas.

The `\abbreviations` command allows to give a list of abbreviations used. It defines a sub-command `\abbrev` with two arguments, of which the first is the abbreviation and the second the meaning. The exact layout will once again depend on the used stylefile.

Here is the example from the beginning of this document:

```
\abbreviations{\abbrev{KAP}{Kluwer Academic Publishers};
\abbrev{compuscript}{Electronically submitted article}}
```

The command `\nomenclature` is closely related to `\abbreviations`. It takes one argument subdivided into `\nomen` parts. The above example gives the correct syntax. For both environments it is legal to have text outside of `\abbrev` respectively `\nomen`, in which case it will just be typeset as flushed text, and you have to provide additional formatting yourself. This is discouraged.

The command `\classification` takes two arguments, the first describing the type of the classification, the second the classification proper. An example might be:

```
\classification{JEL codes}{D24, L60, 047}
```

3.5 Date By using the `\date` command, it is possible to specify reception and acceptance dates. Since this is something that the journal's editor fills in, the default value when no `\date` command is given is to insert a template line that allows easy fill-in of this information. As a result, you are advised not to use this command.

3.6 Motto When you want to specify a piece of poetry / a quotation to specify a motto to your article, you can use the `motto` environment. You can differentiate between prose and poetry. If you are using prose, give an extra option to the environment: `[prose]`

Here ends the description of the opening environment.

4 Mathematical constructs

This section gives an overview of the mathematical constructs that are supported by the class file.

4.1 Symbols As already stated above, the class file includes both `amssymb` and `wasysym` on startup. That is, provided these can be found. There will be no error if they are not available.

All of the commands for symbols from these packages will of course be available. Additionally, there are some commands defined in the `klumac.sty`. A short overview is given in Table 1.²

Please see the documentation for `amssymb` and `wasysym` for the additional symbols these packages provide (most of `amssymb` is also given in (Goossens *et al.*, 1994)).

4.2 Varequation environment Since it can be quite hard to live with the automatic numbering of equations that \LaTeX implements, there is an extra environment defined to handle 'special' equation numbers.

The `varequation` environment allows you to give a 'number' to displayed equations, but without incrementing any \LaTeX counters. If you intend to use mathematical symbols inside the argument (as demonstrated below), be sure to use `\ensuremath`, or otherwise your input may fail under some internal stylefiles (i.e. those that use equation numbers on the left side of the page).

² Some of these commands actually use the AMS fonts. This means that if you do not have these fonts, the commands will fail with a 'missing font.' error.

Table 1. Additional math symbols made available through `kluwer.cls`

I	<code>\I\$</code>	a	<code>\down{a}\$</code>
Res	<code>\Res{a}\$</code>	\uparrow	<code>\dres\$</code>
$\overset{a}{\circ}$	<code>\bigo{a}\$</code>	\Vdash	<code>\forces\$</code>
$\widehat{}$	<code>\concat\$</code>	\leq	<code>\griolo\$</code>
C	<code>\dC\$</code>	\approx	<code>\gsim\$</code>
IE	<code>\dE\$</code>	I	<code>\iduaal\$</code>
IF	<code>\dF\$</code>	lh	<code>\lh\$</code>
IL	<code>\dL\$</code>	\geq	<code>\logr\$</code>
IN	<code>\dN\$</code>	\lesssim	<code>\lsim\$</code>
IP	<code>\dP\$</code>	\downarrow^a	<code>\mapdown{a}\$</code>
Q	<code>\dQ\$</code>	\xrightarrow{a}	<code>\mapright{a}\$</code>
IR	<code>\dR\$</code>	\neq	<code>\nequiv\$</code>
Z	<code>\dZ\$</code>	\otimes	<code>\oast\$</code>
depth	<code>\depth\$</code>	range	<code>\range\$</code>
\diamond	<code>\diamond\$</code>	\uparrow	<code>\res\$</code>
dom	<code>\dom\$</code>	\downarrow	<code>\restrict\$</code>

```
\begin{varequation}{\ensuremath{3\alpha}}
E = mc^2
\end{varequation}
```

And the resulting formula:

$$E = mc^2 \quad (3\alpha)$$

4.3 Subequation environment A subequation environment is available to create equations with sub-numbering of the equation counter. It takes one (optional) argument, that specifies the way that the added counter should be typeset.

```
\begin{subequation}[options]
\begin{equation}
...
\end{equation}
\end{subequation}
```

Possible options are:

`alph` (1a) This is the default.
`Alph` (1A) Uppercase
`arabic` (1.1) Arabic numbers.

4.4 Displayed sentences: theorems and such The `kluwer` class file contains a pre-defined lay-out for theorems, corollaries, etc. These environments have to be defined with the help of L^AT_EX's `\newtheorem` command. Here is an example:

THEOREM 1. *This is a `\newtheorem`.*

The class file also defines a `\newdisplay` command, which is identical to `\newtheorem` in usage, but will give a different layout. Here is an example:

Theorem 1. This is a `\newdisplay`.

For more information on the use of the `\newtheorem` command, please refer to the L^AT_EX manual.

Also, the class defines an extra command, `\newproof`, that can be used for displayed text of a lower level. The major differences are that this command does not define a counter to go with the environment, nor does it have an option to specify such a counter. In fact, it is quite similar to `\newenvironment` in its purpose:

```
\newenvironment{rem}{\par\bigskip \bf Remark: \it}
                {\par\bigskip}
```

is almost identical to³

```
\newproof{rem}{Remark}
```

4.5 Proofs For proofs, you can use the `pf` environment. This will generate the heading ‘*Proof*.’ If you wish to create your own heading, you can use `pf*` and give the `\begin{pf*}` command your alternative heading as an extra argument. Thus, `\begin{pf*}{Proof (of Lemma 3)}` will generate the heading ‘*Proof (of Lemma 3)*.’

5 Lists and displayed text

The document class redefines `itemize` and `enumerate` so that they take an optional argument that makes some parameterization possible.

Also, there are some environments available to do displayed text, for instance, quotations.

5.1 itemize In the case of `itemize` the optional argument specifies the label that should be used as default.

```
\begin{itemize}[$\bullet$]
\item This is an example of a list with an optional argument
\item[--] This is the second item
\end{itemize}
```

```
\begin{itemize}
\item Ordinary.
\end{itemize}
```

Will result in:

- This is an example of a list with an optional argument
- This is the second item

5.2 enumerate In the case of `enumerate` the optional argument specifies the width of the left margin (in approximately the same way as for the `thebibliography` environment).

```
\begin{enumerate}[000000]
\item See the wide left margin below?\\ Next line
```

³ There is more to this, of course, but this is the general idea.


```
\end{enumerate}
\begin{enumerate}
\item Normal.\ \ Next line
\end{enumerate}
```

Will result in:

1. See the wide left margin below?

Next line

1. Normal.

Next line

The default value for the optional argument is 00, which is approximately 1em.

5.3 Using poetry, quotes, and quotations To do ‘ordinary’ quotes and quotations, there are the environments `verse`, `quote` and `quotation`. Here are some small examples of each.⁴

Here’s a verse:

```
I think you will like them,
when you shall see them on a beautiful quarto page,
RICHARD BRINSLEY SHERIDAN, The School for Scandal (1777)
```

A quote:

```
When a proof has been sent me with two or three lines so widely spaced as to make a
grey band across the page, I have often rewritten the passage so as to fill up the lines
better
```

GEORGE BERNARD SHAW, in *The Dolphin* (1940)

A quotation:

```
There was things which he stretched, but mainly he told the truth.
MARK TWAIN, Huckleberry Finn (1884)
```

5.4 Vitae for authors Because of the fact that a lot of our articles end with a short `vita(e)` of the author(s), we have provided a `vitae` environment. Usage is as follows:

```
\begin{vitae}
\Vauthor{Author1} ...
\Vauthor{Author2} ...
\end{vitae}
```

This environment is smart enough to notice the difference between one author and more than one author, which will be reflected in the generated heading.

5.5 Special sections for notes and acknowledgements Just as most articles have a bibliography, a lot of articles also contain ‘Notes’ and ‘Acknowledgements’. For this reason, there are respectively the environments `notes` and `acknowledgements`.

Here are some examples again:

```
\begin{notes}
\item Please note that this class file is provided
{\it as is}, and is \textcopyright\index{copyright}
by Kluwer Academic Publishers. You are free to use
this class file as you see fit, provided that you do
not make changes to the included macro files.
If you do make changes, you are required to rename
the changed files.\index{making changes}
\end{notes}
```

⁴ If you do not know how to use these environments, see the examples in this file.

Notes

1. Please note that this class file is provided *as is*, and is © by Kluwer Academic Publishers. You are free to use this class file as you see fit, provided that you do not make changes to the included macro files. If you do make changes, you are required to rename the changed files.

```
\begin{acknowledgements}
...
\end{acknowledgements}
```

Acknowledgements

I would like to thank Donald E. Knuth for the fact that he wrote this brilliant program, thereby indirectly supporting my wife, my cats and myself.

6 Footnotes and endnotes

When using the `kluwer` class file, you can produce endnotes analogous to \LaTeX footnotes. Instead of the `\footnote` command you use `\endnote`. The command `\theendnotes` should be used to place the endnotes in the text. They will be put in a separate ‘Notes’ section in the `\footnotesize` font.

For use in locations where `\writes fail` (tables etc.), there are the companion commands `\endnotemark` and `\endnotetext`.¹

7 Floats

The `kluwer` class file defines three major kinds of float, and some derivatives like `star-ed` environments and `sub...` versions.

7.1 Tables In general, here is how the `table` environment is intended to be used:

```
\begin{table}[placement options]
...
\caption[optional short version]{Caption text.}
\label{unique label}
\end{table}
```

For *placement options* you can use a variable subset of the following characters:

- t This indicates placement at the top of a page. This may be the current page, but it could also be one of the subsequent ones.
- b This indicates placement at the bottom of a page. This may be the current page, but it could also be one of the subsequent ones.
- p This indicates placement on a page that contains floats only, no text.
- h This should mean ‘here’, but it is not always possible (there may not be enough space left on the page). In that case, \LaTeX will change this to `[ht]`.

The following option is special:

- H Also means ‘here’, but in a more rigorous way. If there is not enough space left on the page, the page will be filled up with whitespace and the float will appear at the top of the next page. The difference between `h` and `H` is that `H` *never* skips over the following text.

The table caption is given with the `\caption` command. This command is required! If you do not have a caption, use `\caption{}`. As you will see in the examples below, the

Table 2. Caption text.

Description 1	Description 2	Description
Row 1, Col 1	Row 1, Col 2	Row 1, Col 3
Row 2, Col 1	Row 2, Col 2	Row 2, Col 3

correct way to tag a table for referencing through `\ref` is by putting the `\label` *after* the caption command.

Examples Two examples of typical tables follow. Their output can be seen in Tables 2 and 3.

We wish to avoid using both vertical rules between columns and horizontal rules between rows. You should, however, always use the following three horizontal rules: a top rule, a middle rule and a bottom rule.

As you can see, we have made some adjustments to the line spacing, effectively disabling vertical rules completely because they create gaps. We saved the old environment in `TABULAR` in case somebody absolutely needs it.

```
\begin{table}
\caption{Caption text.}\label{tab1}
\begin{tabular}{l1l1} \hline
Description 1 & Description 2 & Description \\ \hline
Row 1, Col 1 & Row 1, Col 2 & Row 1, Col 3 \\
Row 2, Col 1 & Row 2, Col 2 & Row 2, Col 3 \\
\hline
\end{tabular}
\end{table}
```

Note in the following example that we also adjusted `tabular*`, so that now the columns always fill the total width you specified. The special `\maxfloatwidth` dimension is the maximum allowed width of the float. This is important because some of our journals have indented floats, so the float is not allowed to be as wide as `\textwidth`, while rotated tables (see below) have an allowed width of (almost) `\textheight`.

```
\begin{table}
\caption{Caption text.}\label{tab2}
\begin{tabular*}{\maxfloatwidth}{l1l1} \hline
Description 1 & Description 2 & Description \\ \hline
Row 1, Col 1 & Row 1, Col 2 & Row 1, Col 3 \\
Row 2, Col 1 & Row 2, Col 2 & Row 2, Col 3 \\
\hline
\end{tabular*}
\end{table}
```

7.2 Additional commands for 'tabular' and 'tabular*' As you saw above, there are some changes to the `tabular` and `tabular*` environments. Apart from the extended baseline and the filling out of `tabular*`, there are also four extra commands to draw partial horizontal lines, these are:

Table 3. Caption text.

Description 1	Description 2	Description
Row 1, Col 1	Row 1, Col 2	Row 1, Col 3
Row 2, Col 1	Row 2, Col 2	Row 2, Col 3

`\rcline`; which leaves some space to the *left* of the line
`\lcline`; which leaves space to the *right* of the line
`\rlcline` and `\lrcline`; these two leave space on both sides.⁵
 These commands are used in the tabular below:

Description 1	Description 2	Description
A	Row 1, Col 2	Row 1, Col 3
B	Row 2, Col 2	Row 2, Col 3

```
\begin{tabular}{l|l|l} \hline
Description 1 & Description 2 & Description \\
\lcline{1-1}\rlcline{2-2}\rceline{3-3}
A& Row 1, Col 2 & Row 1, Col 3 \\
B& Row 2, Col 2 & Row 2, Col 3 \\ \hline
\end{tabular}
```

7.3 Figures For figures, there are the environments `figure` and `figure*`. Both of these support the same optional placement specifiers as `table`; if you want to look into them again, see 105 in section 7.1

A typical figure will look like the following example:

```
\begin{figure}
\centerline{\epsfig{file=tiger.eps,width=16pc}}
\caption{This is the caption of the figure.}
\end{figure}
```

Note that this example needs the line

```
\usepackage{epsfig}
```

in the preamble of your document. Please see the documentation for `epsfig` (or (Goossens *et al.*, 1994)) about the options you can specify in the `\epsfig` command.

7.4 Further options for tables and figures `figure` and `table` also allow you to use the environments `subtable` and `subfigure`. These two environments are analogous to subequation, that was explained previously in section 4.3.

To rotate a table or figure over 90 degrees counterclockwise, enclose the entire table/figure in the `kaprotate` environment.

```
\begin{kaprotate}
\begin{table}
...
\end{table}
\end{kaprotate}
```

⁵ Both mean the same thing, but the double definition maintains symmetry.



Figure 3. This is the caption of the figure. This is the tiger from the GhostScript distribution.

If you rotate a float, it will always use an entire page.

In two-column styles, there is a difference between the starred forms `figure*` and `table*` compared to their normal forms. The starred forms of `table` and `figure` will be doublecolumn, while the ordinary forms are one column. This distinction also holds for rotated versions.

8 References

References are inserted in the `thebibliography` environment. We distinguish two styles of References lists: alphabetically sorted (explained below in section 8.1) and numbered (explained in section 8.2). You can select one of these styles by including the appropriate document option in the `\documentclass` command. *Please refer to the style of the journal concerned when choosing an option for your references.* If no document option is included, the class file will usually presume named references.

8.1 Alphabetically Sorted References For alphabetically sorted references one should include the document option `namedreferences` in the preamble command:

```
\documentclass[namedreferences]{kluwer}
```

Every `\bibitem` in the `thebibliography` environment should have a `\citeauthoryear` command as an option and a key to be referred to. The `\citeauthoryear` should have two arguments: the author's name(s) and the year. The `\citeauthoryear` should be preceded by a `\protect` and in the case of more than two authors the argument with the author's name(s) should contain the first author's name with an *et al.*, e.g.:

```
\bibitem[\protect\citeauthoryear{authorname(s)}{year}]{key}
```

The first part of your References section could look like this:

```
\begin{thebibliography}{} % Note the empty pair of curly braces!

\bibitem[\protect\citeauthor{Smith et al.}{1992}]{Smith92}
Smith, I.N., Johnes, R.S. and Hines, W.P.: 1992,
'Title of the Article',
{\it Journal\} {\bf Vol.~no.~X},
pp.~00--00
```

The following `\cite` commands can now be used in your article: `\cite`, `\shortcite`, `\citeauthor`, `\citeyear`, `\inlinecite` and `\opencite`. For the sample `\bibitem` above, the four cite commands would generate the following reference in your text:

Call:	Produces:
<code>\cite{Smith92}</code>	(Smith et al., 1992)
<code>\citeauthor{Smith92}</code>	Smith et al.
<code>\citeyear{Smith92}</code>	1992
<code>\shortcite{Smith92}</code>	(1992)
<code>\inlinecite{Smith92}</code>	Smith et al. (1992)
<code>\opencite{Smith92}</code>	Smith et al., 1992

With these commands or combinations of them you can create any reference in your text according to the reference style of the journal in which your article is to be published.

If you don't want to make use of references, but just want to generate a bibliographic list at the end of your article, you can use empty square brackets and braces, like this: `\bibitem[]{}.`

You must do the formatting of the bibitems yourself. For the layout conventions we refer you to the authors instructions of the journal in which your article is to be published.

8.2 Numbered References For numbered references one should include the `documentclass` option `numreferences` in the preamble command:

```
\documentclass[numreferences]{kluwer}
```

You should start the `thebibliography` environment with the command:

```
\begin{thebibliography}{99}
```

The second pair of curly braces should contain a number that corresponds with the widest number in the reference list below (usually 99 is the correct value).

You should label every `\bibitem` by putting a key between the curly braces, e.g.: `\bibitem{Smith92}` In your article you can refer to this `\bibitem` with `\cite{Smith92}`.

Using the optional argument of `\bibitem`

Do not use the optional argument to `\bibitem` in this case, since it will produce unexpected results.

The `\cite` command will produce the number of the reference in brackets, e.g. [2]. The `\shortcite`, `\citeauthor` ... commands, as mentioned in Section 8.1, can also be used but they will produce the same output as `\cite`.

You must do the formatting of the bibitems yourself. For the layout conventions we refer you to the authors' instructions of the journal in which your article is to be published.

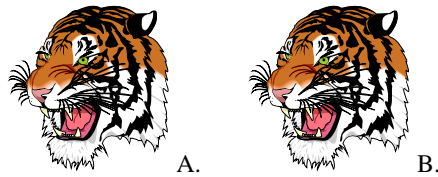


Figure 4. A) This is a figure. B) Second subfigure

8.3 Bib_TE_X You can of course use Bib_TE_X, but if you do so, please make sure you send all the necessary files with the L_AT_EX source file, i.e. the `bb1` file. You could also include the Bib_TE_X output directly in your article. Please don't send us `.bib` files.

Kluwer has Bib_TE_X style files available for the `namedreferences` and `numreferences` options. For named references, the file is called `klunamed.bst` and for `numreferences`, it is called `klunum.bst`. Both files are included with the distribution of the class file.

We refer you to the L_AT_EX and Bib_TE_X manuals for further information on using Bib_TE_X.

9 Special options and extra information

9.1 My pages come out raggedbottom! The sectioning commands always try to keep a certain amount of lines with them on the same page. Values may differ, but the typical values are: 3 lines for section and subsection, 2 lines for subsubsection.

If you find that your pages come out unbalanced, this could be the problem. The other likely problem is that you used a `Here` float on the page, and it didn't fit.

9.2 Authors' instructions The authors' instructions for certain journals are not up-to-date. Quite often the instructions indicate that submissions should be made with double line spacing, or that the journal does not accept electronic figures. Both of these statements do not apply to L_AT_EX submissions. You do not have to change the baselines and we are always quite happy with Encapsulated Postscript figures.

9.3 Side by side figures The use of `minipages` to put two partial floats next to each other, as demonstrated in Goossens *et al.*, 1994, doesn't work with our class files. This is partially because of the way we have implemented floats, and partially a design decision. If you need two figures next to each other, do something like this:

```
\begin{figure}
\centerline{
\epsfig{file=tiger.eps,width=5pc}~A.\quad
\epsfig{file=tiger.eps,width=5pc}~B.
}
\caption{A) This is a figure. B) Second subfigure}
\end{figure}
```

9.4 Double abstracts Sorry, but you cannot have multiple abstracts in one article. If you need one, you have to make a separation in *one* abstract environment.

9.5 Letterspaced text, problems with sections and captions Certain journals require certain headings in a letterspaced font. While the stylefile provides for this, it interferes with writing to auxiliary files like the Table of Contents files. If you encounter incomprehensible error messages, the first thing you should try is to replace the offending `\caption{}` or `\section{}` command by `\caption[]{}` or `\section[]{}`. You will sometimes also have to delete the `\jobname.aux` file.

If you tried this, and it didn't work, you can ask support from `texhelp@wkap.nl`. Please provide us with a log file that contains the error you received.

9.6 Optaining extra fonts / macro packages You may want to get extra packages and/or fonts (like the `wasy` symbol font) if those are not provided at your site.

All \TeX related files on the internet are available for a large central archive called CTAN. Below you will find a list of the main sites and mirrors for CTAN. If you decide to download, please select a site near you (network-wise).

CTAN main archives

```
ftp.dante.de (Deutschland) /tex-archive
tug2.cs.umb.edu (Texas, USA) /tex-archive
ftp.tex.ac.uk (England) /tex-archive
```

CTAN mirrors

```
dongpo.math.ncu.edu.tw (Taiwan) /tex-archive
ftp.center.osaka-u.ac.jp (Japan) /CTAN
ftp.ccu.edu.tw (Taiwan) /pub/tex
ftp.cs.rmit.edu.au (Australia) /tex-archive
ftp.loria.fr (France) /pub/unix/tex/ctan
ftp.muni.cz (The Czech Republic) /pub/tex/CTAN
ftp.riken.go.jp (Japan) /pub/tex-archive
ftp.uni-stuttgart.de (Deutschland) /tex-archive (/pub/tex)
kadri.ut.ee (Estonia) /pub/tex
src.doc.ic.ac.uk (England) /packages/tex/uk-tex
sunsite.unc.edu (North Carolina, USA) /pub/packages/TeX
wuarchive.wustl.edu (Missouri, USA) /packages/TeX
```

If you do not have internet access, some of the packages from these sites can also be obtained by request from KAP, using the address at the end of this document.

9.7 Submitting bug reports and requests If you have questions regarding our class file, or if you have discovered a bug, or if you have a request for something you would like to see included in the next version of this class file, send you comments to: `texhelp@wkap.nl` (e-mail) or to KAP PrePress, Achterom 119, 3311 KB Dordrecht, The Netherlands (regular mail).

Notes

¹ Here is an example of an endnote

References

- Goossens, Michel; Mittelbach, Frank; and Samarin, Alexander: *The \LaTeX Companion*. Addison-Wesley, Reading, MA, USA, 1994.
- Knuth, Donald E: *The \TeX book*. Addison-Wesley, Reading, MA, USA, 1984.
- Lamport, Leslie: *\LaTeX – A Document Preparation System – User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.
- Smith, I.N., Johnes, R.S. and Hines, W.P.: 1992, 'Title of the Article', *Journal Title in Italics* **Vol. no. X**, pp. 00–00

Address for correspondence:

KLUWER ACADEMIC PUBLISHERS PrePress Department,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands
e-mail: `TEXHELP@WKAP.NL`
Fax: +31 78 6392500

Bijlage 13

Quark Xpress: een vakantieliefde

Een manual mee op reis

Frans Goddijn

abstract

Hoe prachtig “ons TeX” ook is, in de wijde wereld daarbuiten is het onbekend. De meeste drukkers en grafische ontwerpers weten niet eens wat het is. Pas als je het magische woord “QuarkXpress” laat vallen, tel je voor hen mee. Afgelopen zomer kocht ik een manual van meer dan 600 pagina's en verslond dit in de vakantie. Wat heeft dat pakket dat wij niet hebben?

keywords

QuarkXpress, manual

1 Welke manual?

Ik kocht, op aanraden van Siep Kroonenberg, *THE QUARK XPRESS BOOK* door David Blatner & Bob Weibel, uitgegeven door Peachpit Press, 1994 (ISBN 1-56609-135-7). Het viel me op dat zo'n oud boek, voor computerboeken een verdacht oud boek, nog verkocht kon worden. Dat lijkt wel TeX, dacht ik. Quark is echter nog niet zo heel oud. Het softwarepakket werd geïntroduceerd in 1987, zo blijkt uit het voorwoord van de oprichter van Quark incorporated.

2 Waartoe op aarde?

“This book, like QuarkXpress, is for everyone who wants to create great documents,” schrijven de auteurs aan het begin van het eerste hoofdstuk. Leslie Lamport zegt het in zijn LaTeX manual bescheidener, als hij stelt dat “LaTeX is a system for typesetting documents”. Hij zegt niet: “typesetting *beautiful* documents”. Paul Abrahams zegt dat wel met zoveel woorden in zijn “TeX for the Impatient”: “Donald Knuth's TeX [...] provides nearly everything needed for high-quality typesetting of mathematical notations as well as of ordinary text.” Knuth zelf zegt het in zijn TeXbook helder: “Gentle reader: This is a handbook about TeX, a new typesetting system intended for the creation of beautiful books [...]”

Het gaat dus om het maken van geweldig mooi drukwerk. Die ambitie hebben de makers van Quark alvast gemeen met de maker van TeX. Het platform-onafhankelijke karakter dat we van TeX kennen is bij Quark¹ niet verder gekomen dan versies voor de Apple en Windows.

3 Klein en fijn

Wat omvang betreft is de essentiële Quark-installatie wel weer aandoenlijk ouderwets: al sinds jaar en dag heb je aan een pakje van zeven floppy's genoeg. Iets om bij stil te staan. Talloze tijdschriften in de kiosken van vandaag worden opgemaakt met een softwarepakket dat nog gewoon vanaf een paar floppen kan worden geïnstalleerd. De lezers van veel van die magazines schrijven hun matig vormgegeven briefjes met pakketten die hele cd-roms beslaan!

Er zijn op die standaardinstallatie wel vele uitbreidingen te koop en te krijg. Die mogelijkheid voor Jan en alleman om uitbreidingen te maken lijkt wel weer wat op de pakketten voor TeX, maar het idee dat er (soms heel grote) bedragen voor kunnen worden gevraagd is ons (gelukkig) vreemd

4 Kip of ei: inhoud vs. vorm

In het tweede hoofdstuk, “Document Construction”, stuit ik op een essentieel verschil met de manier van denken die ik met het gebruik van TeX heb aangeleerd. Ik ben gewend te beginnen met het schrijven van een verhaal. De commando's voor opmaak gaan vooral over de functie van de tekstdelen in het betoog. Dat die functies effect gaan hebben op de vormgeving van het document is van later zorg. Bij Quark echter denken ze anders: eerst specificeer je in een *master page* hoe je document eruit gaat zien — hoe hoog, hoe breed, dubbelzijdig of enkelzijdig en dergelijke. Dan volgen beslissingen over zich herhalende paginaonderdelen, bladnummering en de manier waarop tekst moet overvloeien van de diverse vlakken naar andere tekstvlakken. Bij TeX moet je ook wel beginnen met een of andere documentclass, maar dat kun je na het schrijven zo weer veranderen, terwijl Quark waarschuwt dat een bouwsel zonder gedegen fundering later instort als je aan die fundering nog wat wil veranderen. Je kunt met Quark wel verschillende *master pages* maken voor het gebruik op verschillende pagina's van je totale document. Zo kun je maximaal 127 onderling verschillende indelingen maken en toch de tekst van de ene naar de andere pagina laten vloeien.

1. Ik weet dat kenners nooit voluit QuarkXpress zeggen, maar of ik nu Quark moet zeggen of Xpress, daar ben ik nog niet uit.

5 Pagina vergroten

Bij LaTeX kennen we het `enlargethispage` commando om een tekstpagina ietsje te verdiepen, maar met Quark kun je een pagina tijdelijk twee bladen laten beslaan, zodat een illustratie op (een deel van) het linker- en het rechterblad overvloeit. Aha, zo doen ze dat dus in die bladen, denk ik dan. Ook uitvouwpagina's zijn op die manier te produceren en de auteurs geven er tips bij om te voorkomen dat zulke trucs bij het verdere drukproces onnodig kostbaar worden. Daar heb je natuurlijk meer aan als je een wild jongerenmazagine wil maken dan wanneer je een wetenschappelijk artikel of een simpele nieuwsbrief wilt publiceren.

6 User interface

De user interface voor het veranderen van de indeling van de pagina ziet er (windows natuurlijk) voor mij wel aantrekkelijk uit. Je kunt met de muis bepaalde grenslijnen (bijvoorbeeld die van een kolom in twekolomsopmaak) verslepen, maar je kunt de diverse waarden ook in een schermje met getallen veranderen. Een `columnsep` heet bij Quark `gutter` maar je kunt, zoals we dat gewend zijn, de waarden opgeven in punten, millimeters, inches en zelfs picas. Het boek geeft door zijn heldere uitleg zelfs een gemiddelde TeX-gebruiker weer even een beter inzicht over de manier waarop men kan denken over de indeling van een pagina.

7 Page refs

Een techniek voor automatische inhoudsopgave, zoals wij die gewend zijn te kunnen maken, heb ik niet kunnen vinden. Wat ze bij Quark erg geavanceerd vinden is dat je automatisch een regel kunt plaatsen als "lees verder op pagina x" wanneer een artikel niet naar de volgende pagina overvloeit, maar naar een pagina veel verderop zoals je in tijdschriften vaak ziet.

8 Snijlijnen

Snijlijnen zijn voor Quark een vast onderdeel van het programma terwijl wij daarvoor afhankelijk zijn van externe stylefiles en hun specifieke mogelijkheden en beperkingen (hoewel, een pakket als ConTeXt is daarin weer heel krachtig).

9 Je weet niet wat je ziet

Quark is "What You See Is What You get" en de auteurs van het handboek zijn de eerste om toe te geven dat dit voor het tekstverwerken geen voordeel is. Elk teken dat je intikt kan invloed hebben op de indeling, uitvulling en

afbreking van tekst verderop, op de pagina en op vele pagina's daarna. Het programma gaat dit allemaal meteen zitten uitrekenen en wijzigen, maar intussen heb je als gebruiker weer een nieuw woord ingetikt... je kunt wordafbreking tijdelijk uitzetten, dat scheelt al wat, maar het is raadzaam een andere editor te gebruiken voor het schrijven van de tekst en Quark alleen te gebruiken voor het laatste werk aan redactie en vormgeving. Als alternatieve editor wordt MS WORD aangeraden — van de regen in de drup, zou je denken. Maar wie in WORD consequent gebruik maakt van *style sheets*, zou deze moeten kunnen migreren naar Quark. In Quark zijn de mogelijkheden voor zoeken en vervangen van bepaalde tekstelementen behoorlijk uitgebreid, maar voor een TeX gebruiker met een goeie editor niet echt verbluffend.

10 Over typografie

Het hoofdstuk **Type and Typography** is zeer voedzame kost voor iedereen die met het vormgeven van tekst bezig is. TeX users die ervan zijn overtuigd dat ze vanzelf superieur drukwerk afleveren, domweg "omdat het van TeX komt", zullen wel wat herkennen van de arrogantie van Quark users die schrijven dat "no other desktop-publishing application handles typesetting as powerfully as QuarkXpress". In een dikke honderd pagina's wordt echter heel wat duidelijk gemaakt over kerning (het opschuiven van twee letters ten opzichte van elkaar), afbreking, uitlijnen en regelhoogte. Ook wordt er veel uitgelegd over ADOBE TYPE MANAGER, TRUETYPE, TYPE 1 FONTS, POSTSCRIPT, stuk voor stuk onderwerpen waarover ook de gemiddelde TeX user wel wat "algemene ontwikkeling" kan gebruiken.

Quark biedt de mogelijkheden om vrij simpel in te grijpen in behoorlijk technische aspecten als kerning en tracking (kerning tussen meerdere reeksen letters), waarmee er een extra hulpmiddel is om typografische doodzonden als "weduwen en wezen" te voorkomen. Is er bijvoorbeeld op het laatst in de bewerking net een woord teveel voor een pagina, dan kan een onzichtbaar minieme tracking alle letters net ietsje dichter op elkaar krijgen, zodat het wel past, of de regellengte kan voor een pagina een half procent worden vergroot. Misschien kan dat bij TeX ook, maar dan moet je wel een guru bij de hand hebben.

11 Ligaturen

Bijna zou ik van pure bewondering van mijn stoel vallen, als er een zwakke, om niet te zeggen rotte plek van Quark tevoorschijn komt: ligaturen. Een TeX gebruiker is gewend om fraaie letterparen te zien als *fl*, *fi* en *ffi* in de woorden *efficënt*, *financieel* en *floret*, maar in Quark wordt dat moeilijk. De ligaturen komen niet vanzelf, je moet ze handmatig uit een "expert set" van het gewenste font halen. De spell

checker van Quark kan er niet goed wijs uit en er komen foute afbrekingen van. Ik vraag me af, als het zo moeilijk is, hoe komt het dan dat TeX dit al zo lang zo gemakkelijk kan doen?

12 Humor

Op een belangrijk ding blijft de Quark manual ver achter bij de meeste TeX manuals die ik ken: humor. Een TeX manual is bijna altijd leuk (en als hij van Knuth komt is het nog erudiet ook) maar de Quark-mensen komen niet verder dan “foreign languages have a different word for everything”, en dat is nog de leukste.

13 Guillemets/mots

In ons BABEL-pakket is weleens een misverstand over de juiste spelling van het woord voor franse aanhalingstekens: guillemet of guillemot, het ene is een aanhalingsteken, het andere een vogeltje. In de BABEL-versie die ik thuis draai, werkt `\guillemet` niet, wel `\guillemot` terwijl dat de naam is van het vogeltje. Grappig genoeg maken de auteurs van de Quark manual dezelfde vergissing door in de tekst te spreken van guillemets terwijl ze in de bijbehorende illustratie het teken een guillemot noemen (ze vinden het een “european” quote sign).

14 Geen wiskunde, wel drop caps

Ze zeggen het eerlijk: “QuarkXpress is the wrong program in which to produce a mathematics textbook.” Als alternatief noemen ze de programma’s MATHTYPE en WORD voor Windows. . . Een simpele breuk plaatsen is in Quark al een groots en meeslepend avontuur. Drop caps, de grote initialen aan het begin van een alinea, zijn in Quark wel weer heel simpel op vele spannende manieren te realiseren.

15 Xpress Tags: conversie?

In Quark kun je een veelheid aan zogeheten *style sheets* maken, die het uiterlijk van de pagina bepalen, zoals dat bij ons door stylefiles en macro’s kan worden gedaan. Via een export naar het Rich Text Format kunnen de diverse eigenschappen naar een word processor als WORD worden overgebracht, zodat de benodigde tekst daar kan worden bewerkt. Er kan ook vanuit Quark worden geëxporteerd naar een eigenaardig formaat, Xpress Tags. De tekst gaat er dan in ASCII uitzien als een bastaard tussen TeX en HTML. Dit zou voor een TeX guru een aardige achteringang kunnen betekenen om TeX documenten te converteren naar Quark. Achterin het boek staat op een paar pagina’s een “complete listing” van alle Xpress Tags, alleen kan ik niet geloven dat die claim waar is. Ik vermoed wel dat het alles is dat de

auteurs van die enge tags willen weten, maar er is ongetwijfeld veel meer dan drie pagina’s aan interne code voor alle features die de honderden pagina’s van het boek beschrijven.

16 Plaatjes en kleur

Over illustraties, kleurgebruik en de laatste stap: het drukwerk, staan taloze wetenswaardigheden in meer dan tweehonderd pagina’s. Ook dit zou weer verplichte kost moeten zijn voor iedereen die serieus met het produceren van documenten bezig denkt te zijn. Het gaat me te ver om hier in details te treden. Mijn conclusie was in elk geval: houd het simpel met plaatjes. Bij elke stap verder om het allemaal mooier te maken is de kans dat je iets verkeerd doet viermaal zo groot.

17 Weer thuis

Na thuiskomst ben ik meteen op zoek gegaan naar een PC waarop Quark draait. Ik heb die gevonden, maar Quark bleek een ware vakantie liefde. Onder de zomerzon was ik enthousiast over de vele mogelijkheden en vertederd door de enkele grote zwaktes, maar thuis moest er vooral gewoon gewerkt en geproduceerd worden.

Ik was de auteurs van de manual wel dankbaar toen ik laatst met een klein groepje freelancers op bezoek was bij de hoofdredactie van een tijdschrift. We werden aan het begin van de dag rondgeleid door de drukkerij waar het luxe kleurentijdschrift werd geproduceerd en het zou mij hebben geduizeld als ik niet tevoren uit de Quark manual een zekere basiskennis had opgedaan. Daar stond in een productielijn achter elkaar het viertal drukpersen opgesteld, elk een kleur (CMYK, cyaan, magenta, yellow en black) met voor de omslag een vijfde kleur in een andere productielijn. Het viel me op dat een van de vellen een miniem streepje wit had tussen twee kleurvlakken. Daar had de techniek gefaald die ervoor moet zorgen dat zulke vlakken elkaar iets overlappen.

18 Conclusie

Het zou mooi zijn als ik met LaTeX een POSTSCRIPT bestand kon maken dat door een drukker kon worden verwerkt alsof het een Quark bestand was. Misschien slaagt iemand er eens in om via RTF of nog directer een conversie te maken tussen beide programma’s. Een voorname reden om Quark te laten liggen is de stevige prijs van enkele duizenden guldens voor zo’n klein stapeltje floppy’s. Als ik volgend jaar moet kiezen tussen Quark of een vakantie, dan ga ik lekker op reis, en ditmaal zonder manual.

CONTEXT voor beginners

Nederlandse
versie

Ton Otten & Hans Hagen

PRAGMA ADE

oktober 1997

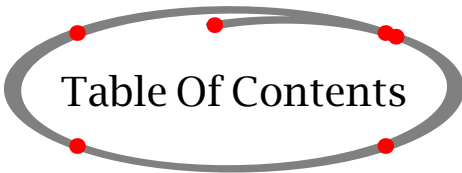
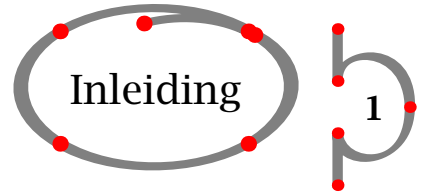


Table Of Contents

Inleiding	2	24 Synoniemen	58
1 Het maken van een document	3	25 Verwijzen naar tekstelementen	59
2 Het verwerken van een file	5	26 Kleur	62
3 Speciale karakters	6	27 Achtergronden bij tekst	63
4 Het definiëren van een document	7	28 Achtergronden op paginavlakken	64
5 Koppen	8	29 Uitlijnen	65
6 Opsommingen	11	30 Interactiviteit in elektronische documenten	66
7 Formules	16	31 Fonts en fontovergangen	70
8 Legenda's	18	32 Samengestelde karakters	75
9 Eenheden	19	33 Paginalayout	76
10 Figuren	21	34 Paragrafen en witruimte	80
11 Tabellen	26	35 Instellingen	84
12 Paragraaf opmaak	36	36 Definiëren van commando's / macro's	85
13 Kolommen	40	37 Diversen	87
14 Voetnoten	42	38 Laden van modules	98
15 Definities	43	A Gebruikersinstellingen	99
16 Genummerde definities	45	B Verwerkingsslagen	99
17 Omlijnde tekst	47	C Hulpfiles	99
18 Omlijnde paragrafen	48	D Commando definities	101
19 Margeteksten	49	E Commando-index	126
20 Pagina's en paginanummering	50	F Onderwerpindex	129
21 Hoofd- en voetteksten	52		
22 Inhoudsopgave (lijsten)	54		
23 Registers	57		



CONTEXT is een document productie systeem en gebaseerd op T_EX. T_EX is een typografische programmeertaal én een programma waarmee u documenten kunt vormgeven en produceren. CONTEXT maakt het werken met T_EX uitermate eenvoudig en stelt u in staat zeer complexe (papieren en elektronische) documenten te vervaardigen.

Deze handleiding beschrijft de mogelijkheden van CONTEXT en de beschikbare commando's en functionaliteit.¹

CONTEXT is ontwikkeld voor en getest in de praktijk: de opmaak en productie van eenvoudige boeken tot zeer geavanceerde technische handleidingen of studieboeken in elektronische of papieren vorm. Deze inleidende handleiding behandelt de CONTEXT functionaliteit, voor zover die van belang is voor de toepassing van standaard tekstelementen in een handleiding of studieboek. CONTEXT kan echter veel meer en voor gebruikers die meer willen zijn andere CONTEXT handleidingen en informatiebronnen beschikbaar.

CONTEXT heeft een meertalige interface, zodat men de eigen taal met CONTEXT kunnen werken. CONTEXT en deze handleiding is beschikbaar in het nederlands, Duits en Engels.

CONTEXT wordt in het public domain uitgezet met behulp van de Nederlandstalige T_EX Gebruikersgroep (NTG). Alle CONTEXT producten en informatie zijn verkrijgbaar op de NTG-server (www.ntg.nl).

¹ Alle papieren en elektronische producten die CONTEXT vergezellen zijn geproduceerd met CONTEXT. Indien mogelijk worden de bronteksten van alle handleidingen elektronisch beschikbaar gesteld. Hierdoor wordt inzicht gegeven in de wijze waarop CONTEXT kan worden gebruikt.



1 | Het maken van een document

Laten we aannemen dat u een eenvoudig document wilt maken. Het heeft enige structuur en bevat een titelpagina, een aantal hoofdstukken en paragrafen. Natuurlijk is er een inhoudsopgave en een index.

CONTEXT maakt een dergelijk document automatisch als u de juiste invoer aanreikt door middel van een file. U dient dus eerst een zogenaamde invoerfile te maken. Een invoerfile bestaat uit een naam en een extensie. U kunt een willekeurige naam kiezen, maar de extensie dient `tex` te zijn. Als een file met de naam `mijnfile.tex` maakt, zult u geen problemen tegenkomen bij het runnen van CONTEXT.

Een invoerfile zou er als volgt uit kunnen zien:

```
\starttekst
\startstandaardopmaak
\regelmidden{Een Document Titel}
\regelmidden{door}
\regelmidden{De Auteur}
\stopstandaardopmaak
\volledigeinhoud
\hoofdstuk{Inleiding}
... uw tekst\index{een indexwoord} ...
\hoofdstuk{Eerste hoofdstuk}
\paragraaf[firstparagraaf]{De eerste paragraaf}
... uw tekst ...
\paragraaf{De tweede paragraaf}
\subparagraaf{de eerste subparagraaf}
... uw tekst\index{nog een indexwoord} ...
\subparagraaf{de tweede subparagraaf}
... uw tekst ...
\paragraaf{De derde paragraaf}
... uw tekst ...
\hoofdstuk{Nog een hoofdstuk}
... uw tekst ...
```

Het maken van een document

```
\hoofdstuk[laatste hoofdstuk]{Het laatste hoofdstuk}
... uw tekst ...
\volledigeindex
\stoptekst
```

CONTEXT verwacht een ASCII invoerfile. Natuurlijk kunt u iedere tekstverwerker gebruiken, maar u dient niet te vergeten dat CONTEXT alleen ASCII invoer kan verwerken. De meeste tekstverwerkers kunnen uw invoerfile exporteren als standaard ASCII ook wel *tekst* genoemd.

De invoerfile bevat de tekst die u wilt zetten en de CONTEXT-commando's. Een CONTEXT-commando begint met een backslash \. Met het commando \starttekst geeft u het begin van de tekst aan. Het gebied voor \starttekst wordt het instelgebied genoemd en wordt gebruikt voor het definiëren van nieuwe commando's en het instellen van de layout van uw document.

Een commando wordt meestal gevolgd door een linker en rechter vierkante haak [] en/of een linker en rechter accolade {}. Het in het voorbeeld gegeven commando \hoofdstuk[laatstchapter]{Het laatste hoofdstuk} vertelt het commando \hoofdstuk aan CONTEXT om een aantal acties uit te voeren ten behoeve van layout, typografie en structuur. Die acties zijn bijvoorbeeld:

1. begin op een nieuwe pagina
2. verhoog het hoofdstuknummer met één
3. plaats het hoofdstuknummer voor de hoofdstuktitel
4. reserveer witruimte na de hoofdstuktitel
5. gebruik een grote letter voor de titel
6. plaats de titel en het nummer in de inhoudsopgave

Deze acties worden uitgevoerd op het argument dat tussen de beide accolades staat: *Het laatste hoofdstuk*.

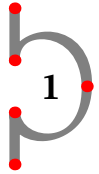
Tot nu toe is nog niet gesproken over [laatste hoofdstuk] dat tussen het commando en de titel staat. Dit is een label met een logische naam dat wordt gebruikt om naar het bewuste hoofdstuk te verwijzen. Dit wordt gedaan met het commando: \in{hoofdstuk}[laatste hoofdstuk].

Nu kan de lijst met acties worden uitgebreid met:

7. geef het label laatste hoofdstuk de waarde van het hoofdstuknummer (en sla dit nummer op voor later gebruik)

Andere acties met betrekking tot voetteksten, resetten van nummering en interactiviteit worden hier nog buiten beschouwing gelaten.

Indien u CONTEXT dit voorbeeld van een invoerfile laat processen, krijgt u een zeer eenvoudig document met een aantal genummerde hoofdstukken en paragrafen, een inhoudsopgave en een register dat twee ingangen bevat.



Tijdens het processen van de invoerfile handelt `CONTEXT` veel zaken af. Een van die zaken is bijvoorbeeld paginanummering. Maar om een inhoudsopgave aan te maken (die meestal aan het begin van het document moet worden geplaatst) heeft `CONTEXT` in eerste instantie te weinig informatie. De invoerfile moet daarom twee maal worden geprocessed (twee verwerkingslagen).

`CONTEXT` produceert een aantal hulpfiles om bepaalde informatie op te slaan. In een enkel geval moet de invoerfile drie maal worden verwerkt door `CONTEXT`.



2 | Het verwerken van een file

`CONTEXT` kan een invoerfile verwerken na een succesvolle installatie. Zie bijlage ?? voor meer informatie.

Als u een invoerfile wilt verwerken met `CONTEXT` typt u na de prompt:

```
ctx filenaam
```

Als uw filenaam `mijnfile.tex` is, wordt dit:

```
ctx mijnfile
```

de extensie `.tex` hoeft niet te worden ingetypt.

Nadat Enter is ingevoerd, wordt de verwerking gestart. `CONTEXT` geeft informatie over de verwerkingsstappen op het beeldscherm. Als de verwerking succesvol is verlopen, verschijnt de prompt en heeft `CONTEXT` een zogenaamde `dvi`-file aangemaakt.

Indien de verwerking niet goed verloopt, bijvoorbeeld omdat u `\stptekst` in plaats van `\stopstekst` heeft ingetypt, geeft `CONTEXT` een `?` op het beeldscherm en geeft aan dat er een fout commando is verwerkt. Er wordt een indicatie gegeven van het type fout en bovendien wordt het regelnummer aangegeven.

Achter het door `CONTEXT` geproduceerde vraagteken kunt u de volgende invoer geven:

H voor helpinformatie over de fout

I voor het invoeren van het correcte `CONTEXT` commando

Q voor het overgaan op batchverwerking

X voor het stoppen van de verwerking

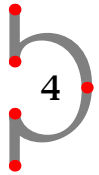
Enter voor het negeren van de fout

Meestal is Enter de beste optie en de verwerking zal gewoon doorgaan. Op het moment dat de verwerking is afgelopen, kunt u de fout herstellen met behulp van uw tekstverwerker.

Sommige fouten zijn aanleiding voor `CONTEXT` om een * op het beeldscherm te genereren en de verwerking te stoppen. Deze fout wordt veroorzaakt door een *fatal error* in de invoerfile. Deze fout kan niet genegeerd worden en de enige optie die u heeft is het typen van `\stop` (in ernstige gevallen kunt u ook `Ctrl-Z` typen). Het programma wordt gestopt en u kunt de fout herstellen.

Tijdens de verwerking informeert `CONTEXT` de gebruiker over de acties die op het document (invoerfile) worden uitgevoerd. `CONTEXT` toont bijvoorbeeld de paginanummers en hoofdstuk- en paragraaftitels op het scherm. Bovendien worden waarschuwingen gegeven. Waarschuwingen hebben meestal een typografische karakter en geven bijvoorbeeld aan dat het afbreken van bepaalde woorden niet of niet goed verloopt. Alle informatie over de verwerking wordt opgeslagen in een log-file waarin de fouten en waarschuwingen en de bijbehorende regelnummers nog eens kunnen worden geraadpleegd.

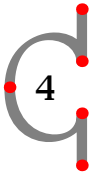
Indien de verwerking succesvol is verlopen heeft `CONTEXT` een nieuwe file aangemaakt met de extensie `.dvi`. In het geval van het verwerken van `mijnfile.tex` heet de file `mijnfile.dvi`. De afkorting `dvi` staat voor *device independent*. Dit betekent dat de file een min of meer onafhankelijk formaat heeft en met behulp van printerdrivers kan worden omgezet in een file die kan worden afgedrukt. De `dvi`-file kan worden omgezet naar een `ps`-file, `HP`-file of welk formaat dan ook.



3 | Speciale karakters

U heeft gezien dat `CONTEXT` commando's beginnen met een `\` (backslash). Dit betekent dat `\` een speciale betekenis heeft voor `CONTEXT`. Naast `\` zijn er andere karakters die speciale aandacht nodig hebben wanneer ze in getypte of gezette vorm moeten worden weergegeven. Tabel 3.1 geeft een overzicht van deze speciale karakters en de manier waarop ze moeten worden ingevoerd om ze correct te kunnen weergeven in uw document.

Andere speciale karakters hebben een betekenis bij het zetten van wiskundige formules en de meeste kunnen alleen in wiskundige mode² worden gebruikt (zie hoofdstuk 7).



Speciaal karakter	Getypte vorm		Gezette vorm	
	Type	Geeft	Type	Geeft
#	<code>\type{#}</code>	#	<code>\#</code>	#
\$	<code>\type{\$}</code>	\$	<code>\\$</code>	\$
&	<code>\type{&}</code>	&	<code>\&</code>	&
%	<code>\type{%}</code>	%	<code>\%</code>	%

Tabel 3.1 Speciale karakters (1).

Speciale karakters	Getypte vorm		Gezette vorm	
	Type	Geeft	Type	Geeft
+	<code>\type{+}</code>	+	<code>\$+\$</code>	+
-	<code>\type{-}</code>	-	<code>\$-\$</code>	-
=	<code>\type{=}</code>	=	<code>\$=\$</code>	=
<	<code>\type{<}</code>	<	<code>\$<\$</code>	<
>	<code>\type{>}</code>	>	<code>\$>\$</code>	>

Tabel 3.2 Speciale karakters (2).



Ieder document wordt gestart met `\starttekst` en afgesloten met `\stoptekst`. Alle overige invoer vindt plaats tussen deze commando's en `CONTEXT` verwerkt alleen die informatie.

Instellingen die van toepassing zijn op het gehele document worden in het instelgebied voor `\starttekst` gedaan.

```
\stelkorpsin[12pt]
\starttekst
Dit is een eenregelig document.
\stoptekst
```



Koppen

Binnen `\starttekst ... \stoptekst` kan een document worden opgedeeld in drie secties:

1. inleidingen
2. hoofdteksten
3. uitleidingen
4. bijlagen

Deze secties worden gedefinieerd met:

```
\startinleidingen ... \stopinleidingen
\starthoofdteksten ... \stophoofdteksten
\startuitleidingen ... \stopuitleidingen
\startbijlagen ... \stopbijlagen
```

In de in- en uitleidende secties produceert het commando `\hoofdstuk` een niet genummerde titel in de inhoudsopgave. Bovendien kan een afwijkende paginanummering worden ingesteld. Binnen *inleidingen* wordt meestal de inhoudsopgave, lijsten met figuren of tabellen, het voorwoord, het dankwoord enz. geplaatst.

De bijlage sectie is bedoeld voor bijlagen (inderdaad). De koppen worden alfabetisch genummerd. Ook in deze sectie kan gewoon `\hoofdstuk` worden gebruikt.

De secties worden ingesteld met:

```
\stelsectieblokin[...] [..., ...=..., ...]
```

5 | Koppen

De structuur van een document wordt bepaald door zijn koppen, zoals hoofdstukken, paragrafen en subparagrafen. Koppen worden gecreëerd met de commando's die in tabel 5.1 staan:

```
\hoofdstuk[ref,...]{...}
```

Genummerde kop	Ongenummerde kop
<code>\hoofdstuk</code>	<code>\titel</code>
<code>\paragraaf</code>	<code>\onderwerp</code>
<code>\subparagraaf</code>	<code>\subonderwerp</code>
<code>\subsubparagraaf</code>	<code>\subsubonderwerp</code>
...	...

Tabel 5.1 Koppen.

```
\paragraaf[ref,..]{...}
```

```
\subparagraaf[ref,..]{...}
```

```
\titel[ref,..]{...}
```

```
\onderwerp[ref,..]{...}
```

```
\subonderwerp[ref,..]{...}
```

Deze commando's produceren een kop en een kopnummer in een bepaalde grootte en met een vooraf ingestelde verticale witruimte voor en na de kop.

De koppen hebben twee verschijningsvormen. Bijvoorbeeld:

```
\titel[hasselt bij nacht]{Hasselt bij nacht}
```

en

```
\titel{Hasselt bij nacht}
```

De vierkante haken zijn optioneel en worden gebruikt voor interne verwijzingen. Verwijzen doet u vervolgens met: `\op{pagina}[hasselt bij nacht]`.

Natuurlijk kunnen koppen in een door u zelf gedefinieerde vormgeving worden weergegeven. Dit gebeurt met de commando's `\stelkopin` en `\definieerkop`.

```
\definieerkop[.1.][.2.]
```

```
\stelkopin[...][...,...=...,...]
```

```
\definieerkop
  [mijkop]
  [paragraaf]

\stelkopin
  [mijkop]
  [nummerletter=vet,
   tekstletter=kap,
   voor={\haarlijn\pagina[nee]},
   na={\geenwitruimte\pagina[nee]\haarlijn}]

\mijkop[mijkop]{In Hasselt wonen kopstukken}
```

Een nieuwe kop `\mijkop` wordt gedefinieerd en erft daarbij de eigenschappen van `\paragraaf`. Een dergelijke kop ziet er als volgt uit:

5.1 IN HASSELT WONEN KOPSTUKKEN


Een ander commando met betrekking tot koppen is `\stelkoppenin`. U kunt dit commando gebruiken voor het instellen van de nummering van genummerde koppen. Als u typt:

```
\stelkoppenin
  [variant=inmarge,
   scheider=--]
```

zullen alle nummers in de marge verschijnen en worden subnummers als volgt weergegeven: 1-1.

Commando's als `\stelkoppenin` worden bij voorkeur in het instelgebied van uw invoerfile geplaatst.

```
\stelkoppenin[...=...,...]
```



6 | Opsommingen



6

Informatie kan worden gestructureerd met behulp van opsommingen. Er zijn genummerde en ongenummerde opsommingen. Het commando om opsommingen te maken ziet er als volgt uit:

```
\startopsomming[...][...=...,...] ... \stopopsomming
```

Bijvoorbeeld:

```
\startopsomming[R,opelkaar,ruim]
\som Hasselt ontstond in 14e eeuw.
\som Hasselt staat bekend als een Hanzestad.
\som Hasselt's naam is ontleend aan een boom.
\stopopsomming
```

Binnen het `\startopsomming ... \stopopsomming` paar wordt ieder nieuw gegeven gestart met het commando `\som`. De spatie achter `\som` is vereist. In het bovenstaande voorbeeld specificeert `R` dat een Romeinse nummering is gewenst en `opelkaar` zorgt ervoor dat de items zonder witruimte opelkaar worden geplaatst. De instelling `ruim` zorgt voor extra ruimte na de scheider. In gezette vorm ziet het voorbeeld er als volgt uit:

- I. Hasselt ontstond in 14e eeuw.
- II. Hasselt staat bekend als een Hanzestad.
- III. Hasselt's naam is ontleend aan een boom.

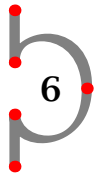
Voor het zetten van opsommingen zijn twee verwerkingsslagen nodig. Dit betekent dat u uw invoerfile twee keer door `CONTEXT` moet laten verwerken om een correcte layout te verkrijgen. Tussen de vierkante haken staat informatie over de itemscheiders en locale instellingen van opsommingen.

U kunt natuurlijk ook een eigen itemscheider definiëren door middel van het commando `\definieersymbool`. Als u bijvoorbeeld het volgende invoert:

```
\definieersymbool[5][\clubsuit$]
\startopsomming[5,opelkaar]
\som Hasselt werd gebouwd op een rivierduin.
\som Hasselt ligt op een kruising van twee rivieren.
\stopopsomming
```

Opsommingen

Argument	Itemscheider
1	–
2	•
3	*
:	:
n	1 2 3 4 ...
a	a b c d ...
A	A B C D ...
r	i ii iii iv ...
R	I II III IV ...



Tabel 6.1 Itemscheiders in opsommingen.

Krijgt u:

- ♣ Hasselt werd gebouwd op een rivierduin.
- ♣ Hasselt ligt op een kruising van twee rivieren.

Soms zijn binnen opsomming koppen gewenst. In die gevallen wordt in plaats van het commando `\som` het commando `\kop` ingevoerd.

Hasselt ligt in Overijssel en er is een aantal gebruiken dat typerend is voor deze provincie.

`\startopsomming`

`\kop kraamschudden`

Na de geboorte van een kind komen de burens de nieuwe ouders bezoeken. De vrouwen bewonderen het kind en de mannen beoordelen het kind (als het een jongen is) of hij een sterke man zal worden. De burens brengen een krentewegge mee. Dat is een krentebrood van ongeveer 1 `\Meter` lengte. Natuurlijk wordt ook op het nieuwe kind geklonken.

`\kop nabuurschap (naberschap)`

In de kleine gemeenschappen waren de mensen vroeger sterk op elkaar aangewezen. Leden van zo'n `{\em nabuurschap}` hielpen elkaar bij het oogsten, begrafenissen of tegenslagen die de gemeenschap te verwerken kreeg.

`\kop Abraham / Sarah`



Opsommingen

Als mensen 50 worden, wordt er van hen gezegd dat ze Abraham of Sarah zien. Het is gewoonte deze mensen een Abraham of Sarah van speculaas te geven.

6

`\stopopsomming`

Het commando `\kop` kan worden ingesteld met `\stelopsommingin`. In geval van een pagina-overgang zal een nieuwe `\kop` altijd aan het begin van de eerstvolgende pagina worden geplaatst.

Het eerdere ingevoerde voorbeeld over oude gebruiken komt er na verwerking als volgt uit te zien:

Hasselt ligt in Overijssel en er is een aantal gebruiken dat typerend is voor deze provincie.

- kraamschudden

Na de geboorte van een kind komen de buren de nieuwe ouders bezoeken. De vrouwen bewonderen het kind en de mannen beoordelen het kind (als het een jongen is) of hij een sterke man zal worden. De buren brengen een krentewegge mee. Dat is een krentebrood van ongeveer 1 m lengte. Natuurlijk wordt ook op het nieuwe kind geklonken.

- nabuurschap (naberschap)

In de kleine gemeenschappen waren de mensen vroeger sterk op elkaar aangewezen. Leden van zo'n *nabuurschap* hielpen elkaar bij het oogsten, begravenissen of tegenslagen die de gemeenschap te verwerken kreeg.

- Abraham / Sarah

Als mensen 50 worden, wordt er van hen gezegd dat ze Abraham of Sarah zien. Het is gewoonte deze mensen een Abraham of Sarah van speculaas te geven.

De mogelijke instellingen zijn weergegeven in tabel 6.2.

U kunt lokale instellingen meegeven tussen de vierkante haken direct achter het commando `\startopsomming`, maar voor de consistentie kunt u de voorkeuren ook voor het gehele document instellen met `\stelopsommingin`.

De instelling `kolommen` wordt altijd gebruikt in combinatie met een geschreven aantal. Indien u typt:

```
\startopsomming[n,kolommen,vier]
\som Achter 't Werk
.
.
.
\som Justitiebastion
\stopopsomming
```

Krijgt u:

Opsommingen

- | | | | |
|-------------------|-------------------|-----------------|---------------------|
| 1. Achter 't Werk | 4. Eikenlaan | 7. Heerengracht | 10. Julianakade |
| 2. Baangracht | 5. Eiland | 8. Hofstraat | 11. Justitiebastion |
| 3. Brouwersgracht | 6. Gasthuisstraat | 9. Hoogstraat | |

Als u een opsomming na een kort intermezzo verder wilt laten lopen dan kan dat. Als u bijvoorbeeld `\startopsomming[verder, kolommen, drie, ruim]` intypt, gaat de nummering verder in drie kolommen.

- | | | |
|------------------|--------------------|----------------------|
| 12. Kaai | 18. Prinsengracht | 24. Vicariehof |
| 13. Kalverstraat | 19. Raamstraat | 25. Vissteeg |
| 14. Kastanjelaan | 20. Ridderstraat | 26. Watersteeg |
| 15. Keppelstraat | 21. Rosmolenstraat | 27. Wilhelminalaan |
| 16. Markt | 22. Royenplein | 28. Ziekenhuisstraat |
| 17. Meestersteeg | 23. Van Nahuijsweg | |

De instelling `ruim` vergroot de horizontale witruimte tussen scheider en itemtekst.

```
\steloopsommingin[.1.][...2,...][...=,...]
```

Een opsomming binnen een opsomming wordt automatisch op de juiste wijze gezet. Als u bijvoorbeeld intypt:

Steden kunnen zelf de hoogte van bepaalde belastingen vaststellen. Hierdoor kunnen de kosten voor gemeentelijke belastingen van stad tot stad verschillen. Die verschillen lopen op tot 50% in belastingen als:

```
\steloopsommingin[2][breedte=5em]
```

```
\startopsomming[n]
```

```
\som de onroerend goed belasting
```

De onroerend goed belasting bestaat uit twee componenten:

```
\startopsomming[a, opelkaar]
```

```
\som het deel voor de eigenaar
```

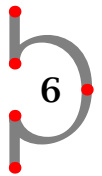
```
\som het deel voor de huurder / bewoner
```

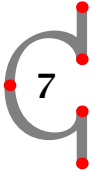
```
\stopopsomming
```

In het geval dat een pand geen huurder heeft, betaalt de eigenaar beide componenten.

```
\som de hondenbelasting
```

De eigenaar van een hond betaalt hondenbelasting. Als hond wordt aangeschaft of doodgaat, dient dat te worden gemeld bij de gemeente.





Set up	Meaning
standaard	standaard instellingen
opelkaar	geen witruimte tussen onderdelen
aaelkaar	geen wit voor en na de opsomming
aansluitend	weinig witruimte na de scheider
ruim	extra witruimte na de scheider
inmarge	scheider in de marge
opmarge	scheider op de marge
afsluiter	afsluiter na de scheider
kolommen	in kolommen
intro	geen pagina-overgang na introductieregel
verder	doornummeren

Tabel 6.2 Instellingen van opsommingen.

`\stopopsomming`

Dan wordt subopsomming automatisch geplaatst en de horizontale witruimte op het tweede niveau ingesteld met `\steloopsommingin[2][breedte=5em]`.

Steden kunnen zelf de hoogte van bepaalde belastingen vaststellen. Hierdoor kunnen de kosten voor gemeentelijke belastingen van stad tot stad verschillen. Die verschillen lopen op tot 50% in belastingen als:

1. de onroerend goed belasting

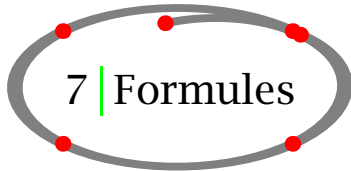
De onroerend goed belasting bestaat uit twee componenten:

- a. het deel voor de eigenaar
- b. het deel voor de huurder / bewoner

In het geval dat een pand geen huurder heeft, betaalt de eigenaar beide componenten.


2. de hondenbelasting

De eigenaar van een hond betaalt hondenbelasting. Als hond wordt aangeschaft of doodgaat, dient dat te worden gemeld bij de gemeente.



7 | Formules

Genummerde formules maakt u met:



```
\plaatsformule[ref,..].1.$$2.$$
```



```
\startformule ... \stopformule
```

Een tweetal voorbeelden:

```
\plaatsformule[formule:eenformule]
\startformule
y=x^2
\stopformule
```

```
\plaatsformule
\startformule
\int_0^1 x^2 dx
\stopformule
```

$$y = x^2 \tag{7.1}$$

$$\int_0^1 x^2 dx \tag{7.2}$$

Tussen `\startformule` en `\stopformule` werkt u in de mathematische mode en kunnen formules worden gedefinieerd met behulp van \TeX -commando's. De volgende boeken worden aangeraden om het typesetten van formules onder de knie te krijgen:

- het *\TeX Book* van Donald E. Knuth.
- \TeX for beginners van ...
- An Introduction to \TeX van ...

De mathematische mode kent twee verschijningsvormen: tekstmode en displaymode. Mathematische expressies in tekstmode worden tussen \$ en \$ geplaatst.

```
De gemeente Hasselt beslaat een gebied van 42,05 \Square
\Kilo \Meter. Bij een denkbeeldige cirkel met de Markt als
```

middelpunt M dan kan de diameter van de cirkel worden berekend met $\frac{1}{4}\pi r^2$.

Dit wordt:

De gemeente Hasselt beslaat een gebied van 42,05 km². Bij een denkbeeldige cirkel met de Markt als middelpunt M dan kan de diameter van de cirkel worden berekend met $\frac{1}{4}\pi r^2$.

De vele $\{ \}$ in $\frac{1}{4}\pi r^2$ zijn essentieel om operaties in de expressie te scheiden. Als de buitenste accolades worden weggelaten: $\frac{1}{4} \pi r^2$, wordt een ongewenst resultaat verkregen: $\frac{1}{4\pi r^2}$.

Displaymode begint met $\$$ en eindigt met $\$$.

Als u typt:

```

 $\int_0^1 x^2 dx$ 

```

krijgt u een expressie die op het midden van de pagina wordt 'gedisplayed'.

$$\int_0^1 x^2 dx$$

Als u de beide $\$$ uit dit voorbeeld vergelijkt met de eerdere voorbeelden dan valt op dat $\$$ en $\backslash\text{startformule} \dots \backslash\text{stopformule}$ een vergelijkbare functie hebben.

Het commando $\backslash\text{plaatsformule}[]$ zorgt voor de witruimte voor en na een formule en zorgt tevens voor het nummeren. De vierkante haken zijn optioneel en wordt gebruikt voor het verwijzen naar de formule en voor het aan- en uitzetten van nummeren.

$$y = x^2 \tag{7.3}$$

$$y = x^3 \tag{7.4}$$

$$y = x^4 \tag{7.5}$$

Formule 7.4 is als volgt ingevoerd:

```

\plaatsformule[middelste]
\startformule
y=x^3
\stopformule

```

Het label $[middelste]$ wordt gebruikt voor het verwijzen naar deze formule. Zo'n referentie wordt opgeroepen met $\backslash\text{in}\{formule\}[middelste]$.

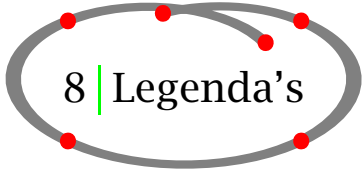
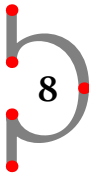
Indien nummering ongewenst is typt u:

`\plaatsformule[-]`

Het nummeren van formules wordt ingesteld met `\stelnummeren`. In deze handleiding is het nummeren ingesteld met `\stelnummeren[wijze=perhoofdstuk]`. Dit betekent dat het hoofdstuknummer voorafgaat aan het formulenummer en dat het nummeren van de formule bij ieder nieuw hoofdstuk wordt gereset. Uit oogpunt van consistentie wordt het nummeren van tabellen, figuren en formules met hetzelfde commando ingesteld. Het commando `\stelnummeren` dient dan ook in het instelgebied van uw invoerfile te staan.

Formules kunnen worden ingesteld met:

```
\stelformulesin[...=...]
```



De commando's `\startlegenda` ... `\stoplegenda` worden gebruikt om de legenda bij een formule in te voeren. Een opgemaakte formule en de bijbehorende legenda kunnen er bijvoorbeeld als volgt uitzien.

$$s = vt + \frac{1}{2}at^2 \tag{8.1}$$

- s = plaats m
- v = snelheid m/s
- t = tijd s
- a = versnelling m/s²

Daarvoor moet u het volgende invoeren:

```
\plaatsformule$$ s = v t + \frac{1}{2} a t^2 $$
\startlegenda
\leg s \ \ plaats      \ \ m      \ \
\leg v \ \ snelheid    \ \ m/s    \ \
\leg t \ \ tijd        \ \ s       \ \
\leg a \ \ versnelling \ \ m/s^2 \ \
\stoplegenda
```

Het commando `\leg` begint een nieuwe rij en `\\` is een kolomscheider. De laatste `\\` in iedere rij is essentieel. De spaties zijn optioneel en hebben geen effect op het gezette resultaat. De commando's worden echter snel verkeerd getypt en daarom is het gebruik van spaties ten behoeve van controle en het voorkomen van fouten geen overbodige luxe. Het advies luidt dan ook: probeer uw invoerfile zo leesbaar mogelijk te maken.

9

```
\startlegenda[.1.]\leg.2.\\.3.\\.4.\\\leg ... \stoplegenda
```

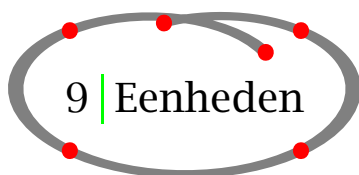
De eerste kolom van `\leg` wordt gebruikt voor symbolen en staat in de mathematische mode. De tweede kolom wordt gebruikt voor uitleg bij de symbolen of expressies en de derde kolom voor eenheden en dimensies. Omdat eenheden en dimensies gebruik maken van sub- en superscript is de derde kolom weer in mathematische mode, maar wordt de tekst gezet in het actuele (niet-mathematische) font.

In aanvulling op `legenda's` is er een vergelijkbaar commando voor het weergeven van gegevens die in een dergelijke formule kunnen worden ingevuld.

Het commando kan er bij het voorgaande voorbeeld als volgt uitzien:

```
\startgegeven
\geg snelheid    \\ v \\ 10~m/s  \\
\geg versnelling \\ a \\ - 3~m/s^2 \\
\geg tijd        \\ t \\ 3~s      \\
\stopgegeven
```

De twee laatste kolommen zijn weer in mathematische mode, waarbij de laatste kolom weer in een niet-mathematische font wordt weergegeven. De `~` zijn hier vereist omdat spaties in de mathematische mode worden genegeerd.



Om consistentie in het gebruik van eenheden en dimensies in een document af te dwingen kunt u uw eigen eenheden definiëren. Eenheden worden gedefinieerd in het instelgebied van uw invoerfile.

`CONTEXT` heeft een eigen module waarin nagenoeg alle SI-eenheden zijn opgenomen. Als de module `\gebruikmodule[eenheid]` geladen is, kunnen eenheden als volgt worden opgeroepen:

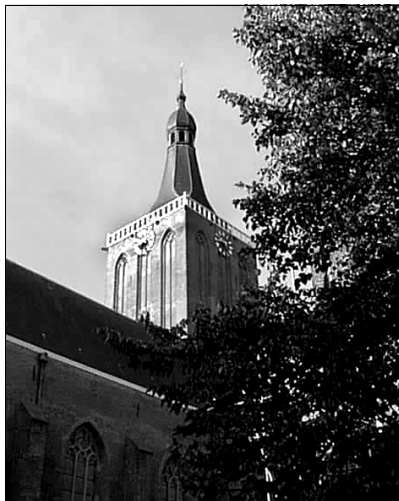
10 | Figuren

10

Foto's en figuren worden in uw document geplaatst met het volgende commando:

```
\plaatsfiguur  
[]  
[fig:kerk]  
{Stephanus Kerk.}  
\naam{hass24g}}
```

Na verwerking wordt dit:



Figuur 10.1 Stephanus Kerk.

Het commando `\plaatsfiguur` zorgt voor nummering en verticale witruimte voor en na het figuur. Bovendien initieert het commando het floatmechanisme. Dit mechanisme zorgt ervoor dat de figuur blijft 'zweven' tot de meest optimale locatie in het document is gevonden. Meestal is dit verplaatsen van een figuur nodig, omdat figuren relatief groot zijn. Het floatmechanisme kan worden beïnvloed binnen het eerste hakenpaar.

Het commando `\plaatsfiguur` is een verbijzondering van het commando:

`\plaatsblok[.1.][ref,..]{.2.}{.3.}`

De mogelijke opties van `\plaatsfiguur` worden beschreven in tabel 10.1.

Optie	Betekenis
hier	plaats figuur bij voorkeur op deze locatie
forceer	negeer het floatmechanisme en plaats figuur
pagina	plaats figuur bovenaan volgende pagina
boven	plaats onderschrift boven de figuur
onder	plaats onderschrift onder de figuur
links	plaats figuur tegen de linkermarge
rechts	plaats figuur tegen de rechtermarge

Tabel 10.1 Opties in `\plaatsfiguur`.

Het tweede paar vierkante haken wordt gebruikt voor verwijzingen naar de figuur. U kunt naar dit figuur verwijzen door te typen:

`\in{figuur}[fig:kerk]`

De eerste accolades van `\plaatsfiguur` worden gebruikt voor de bijschrijft. Op die plaats kan iedere gewenste tekst worden ingevoerd. Indien u geen nummering of bijschrift wenst, typt u `{geen}`. Het label *figuur* wordt ingesteld met `\stelblokkopjes` in en het nummeren wordt ingesteld of hersteld met `\stelnummeren` in (zie paragraaf 37.1).

Het tweede paar accolades wordt gebruikt om externe figuurfiles aan te roepen of het figuur te definiëren.

In het volgende voorbeeld ziet u hoe `[Hasselt]` wordt gedefinieerd binnen het laatste accoladepaar en wordt de functie van `\plaatsfiguur{}{}` toegelicht.

```
\plaatsfiguur
  {Een omlijnd Hasselt.}
  {\omlijnd{\tfd Hasselt}}
```

Dit wordt:

Hasselt

Figuur 10.2 Een omlijnd Hasselt.

Meestal worden figuren opgemaakt of opgewerkt in pakketten als Corel Draw of Illustrator en PhotoShop. De figuren zijn dan beschikbaar als files. In combinatie met `TEXUTIL`

ondersteunt `CONTEXT` het `eps` fileformaat.³ U dient daarom de eventuele `tif`-files om te zetten naar `eps`. Voordat de invoerfile wordt verwerkt, heeft `CONTEXT` informatie nodig over de figuren, zoals fileformaat en dimensies. Deze informatie wordt verkregen door middel van het commando `\gebruikexternfiguur`.

In figuur 10.3 ziet u een foto en een opgemaakt plaatje in één figuur.

10



een bitmap foto



een vector plaatje

Figuur 10.3 Grachten in Hasselt.

U kunt deze figuur als volgt maken:

```
\gebruikexternfiguur
  [foto][hass03g]
  [type=tif,breedte=.4\tekstbreedte]

\gebruikexternfiguur
  [plaatje][gracht]
  [type=eps,breedte=.4\tekstbreedte]

\plaatsfiguur
  [hier,forceer]
  [fig:grachten]
  {Grachten in Hasselt.}
  \startcombinatie[2*1]
  {\foto}{een bitmap foto}
  {\plaatje}{een vector plaatje}
  \stopcombinatie
```

`\gebruikexternfiguur[.1.][.2.][. . . , . . . = . . . , . . .]`

³ Enkele `dvi` naar `ps` drivers ondersteunen ook het `tif` formaat. Deze handleiding is bijvoorbeeld geproduceerd met `DVIPSONE`.

Figuren

Het commando `\gebruikexternfiguur` heeft drie paren met vierkante haken. Het eerste paar bevat de logische naam van het figuur. Deze naam is optioneel en wordt bijvoorbeeld gebruikt als een figuur meer dan eens in een document wordt aangeroepen.

Het tweede paar bevat de filenaam (zonder extensie). Het derde paar geeft ruimte om het fileformaat te definiëren en enkele waarden bij de figuur in te stellen. In goed gestructureerde documenten staat `\gebruikexternfiguur` in het instelgebied van de invoerfile.

```
\startcombinatie[...] ... \stopcombinatie
```

De commando's `\startcombinatie ... \stopcombinatie` worden gebruikt voor het combineren van twee plaatjes in een figuur. Het aantal plaatjes wordt tussen de vierkante haakjes ingetypt. Wanneer de plaatjes onder elkaar moeten staan, typt u `[1*2]`. U kunt zich misschien voorstellen wat er gebeurt als u zes plaatjes combineert met `[3*2]` (`[h*v]`).

De voorbeelden hiervoor zijn eigenlijk al voldoende om goed geïllustreerde documenten te creëren. Soms is het echter wenselijk om figuur en tekst meer met elkaar te combineren. U kunt daarvoor gebruik maken van het commando:

```
\startbloktekst[.1.][ref]{.2.}{.3.} ... \stopblok
```

Het effect van:

```
\startfiguurtekst
[links]
[fig:inwoners]
{geen}
{\externfiguur[hass07g][type=tif,breedte=.5\zetbreedte,kader=aan]}
Het inwonersaantal van Hasselt heeft altijd
gevarieerd met de economische ontwikkelingen. De
Dedemsvaart werd bijvoorbeeld rond 1810 gegraven. Dit
kanaal loopt door Hasselt en de handel floreerde. De
populatie nam binnen 10 jaar toe met ongeveer 40%.
Tegenwoordig heeft de Dedemsvaart geen economische waarde
meer en zijn de grachten slechts een toeristische
attractie.
\stopfiguurtekst
```

wordt in de onderstaande figuur getoond.



Het inwonersaantal van Hasselt heeft altijd gevarieerd met de economische ontwikkelingen. De Dedemsvaart werd bijvoorbeeld rond 1810 gegraven. Dit kanaal loopt door Hasselt en de handel floreerde. De populatie nam binnen 10 jaar toe met ongeveer 40%. Tegenwoordig heeft de Dedemsvaart geen economische waarde meer en zijn de grachten slechts een toeristische attractie.

```
\externfiguur[...][...=...]
```

Het laatste paar accolades bevat het commando `\externfiguur`. Dit commando geeft u de vrijheid om met een figuur te doen wat u wilt. Het commando `\externfiguur` heeft twee paren van vierkante haken. Het eerste paar wordt gebruikt voor de filenaam van de figuur zonder extensie, het tweede voor het fileformaat en de dimensies. Het is niet moeilijk te voorspellen wat er gebeurt als u het volgende intypt:

```
\inmarge
  {\externfiguur
   [hass23g]
   [type=tif,
    breedte=\margebreedte,
    kader=aan]}
```

De opmaak van figuren wordt ingesteld met:

```
\stelplaatsblokkenin[...=...]
```

De nummering en de labels worden ingesteld met:

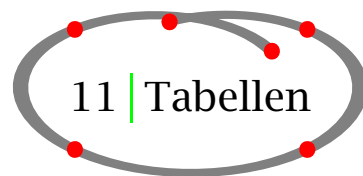
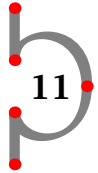
```
\stelblokkopjesin[...=...]
```

De beide commando's worden in het instelgebied van de invoerfile geplaatst en zijn geldig voor het totale document.

```
\stelplaatsblokkenin
  [plaats=rechts]
```

```
\stelblokkopjesin
  [plaats=boven,
   hoogte=.4\zethoogte,
   letter=vetschuin]
\plaatsfiguur
  {Een karakteristiek zicht op Hasselt.}
  {\externfiguur[hass18g][kader=aan]}
```

Figuur 10.4 *Een karakteristiek zicht op Hasselt.*



In het algemeen bestaat een tabel uit een aantal kolommen die onafhankelijk van elkaar zijn gecentreerd of links of rechts zijn uitgelijnd. Kolommen kunnen ook zijn uitgelijnd op de decimale komma. Koppen worden boven een of meerdere kolommen geplaatst en horizontale en verticale lijnen worden geheel of gedeeltelijk over de breedte van de tabel getrokken.

Dit schrijft Michael J. Wichura in zijn inleiding op de handleiding van `TABLE` (`TABLE` manual, 1988). Michael Wichura is tevens de auteur van de `TABLE`-macro's die `CONTEXT` gebruikt om

tabellen te zetten. Er zijn meerdere `CONTEXT`-macro's aan toegevoegd ten behoeve van consistente horizontale spatiering en om het opmaken van tabellen iets inzichtelijker te maken.⁴

Voor het plaatsen van de tabel wordt gebruik gemaakt van `\plaatstabel` dat ook weer een verbijzondering is van:

```
\plaatsblok[.1.][ref,...]{.2.}{.3.}
```

11

Verder wordt voor het plaatsen van de tabel gebruik gemaakt van:

```
\starttabel[...] ... \stoptabel
```

De definitie van een tabel zou er als volgt uit kunnen zien:

```
\plaatstabel[hier][tab:ships]{Schepen die Hasselt aandeden.}
\starttabel[|c|c|]
\HL
\NC \bf Jaar \NC \bf Aantal schepen \NC\SR
\HL
\NC 1645 \NC 450 \NC\FR
\NC 1671 \NC 480 \NC\MR
\NC 1676 \NC 500 \NC\MR
\NC 1695 \NC 930 \NC\LR
\HL
\stoptabel
```

Na verwerking ziet deze tabel er als volgt uit:

Jaar	Aantal schepen
1645	450
1671	480
1676	500
1695	930

Tabel 11.1 Schepen die Hasselt aandeden.

⁴ `CONTEXT` is gemaakt voor niet-technici in het `WYSIWYG`-tijdperk. Daarom is steeds gekozen voor begrijpelijke benamingen van commando's en zijn veel handelingen voorgedefinieerd om het programmeren in `TEX` overbodig te maken. Mede daardoor is ook het opmaken van tabellen iets eenvoudiger geworden.

Tabellen

Het eerste commando `\plaatstabel` heeft dezelfde functie als `\plaatsfiguur`. Het draagt zorg voor verticale spatiering en tabelnummering. Het floatmechanisme wordt geïnitieerd en de tabel komt uiteindelijk op de meest optimale plaats in het document terecht.

De tabelinhoud wordt tussen het `\starttabel` \dots `\stoptabel` paar ingevoerd. Tussen vierkante haken achter `\starttabel` wordt het tabelformaat gedefinieerd met formaataanduidingen en gescheiden door kolomscheiders `|` (zie tabel 11.2).

Aanduiding	Betekenis
	kolomscheider
c	centreer
l	links uitlijnen
r	rechts uitlijnen
s<n>	spatie tussen kolommen op waarde $n = 0, 1, 2$
w<>	kolombreedte

Tabel 11.2 Tabel formaataanduidingen.

In aanvulling op formaataanduidingen zijn er formaatcommando's. Tabel 11.3 toont een aantal essentiële commando's.

Commando	Betekenis
<code>\JustLeft</code>	links uitlijnen en negeer kolomindeling
<code>\JustRight</code>	rechts uitlijnen en negeer kolomindeling
<code>\JustCenter</code>	centreer en negeer kolomindeling
<code>\SetTableToWidth{}</code>	specificeer tabelbreedte
<code>\use{n}</code>	gebruik de ruimte over van n kolommen

Tabel 11.3 Tabel formaatcommandos.

In de voorbeelden die u tot nu toe heeft gezien zijn een aantal `CONTEXT` tabelcommando's toegepast. Deze commando's zijn ietwat langer dan de originele commando's, maar zijn minder cryptisch en dragen zorg voor veel tabeltypografie. In tabel 11.4 is een overzicht gegeven van deze commando's.

De tabellen en hun definitie worden hieronder getoond. Voor meer uitleg en voorbeelden van geavanceerde tabellen kunt u de `TABLE` manual van M.J. Wichura raadplegen.

```
\plaatstabel  
[hier, forceer]
```


Tabellen

Commando		Betekenis
<code>\NR</code>	next row	maak rij, geen aanpassing verticale witruimte
<code>\FR</code>	first row	maak rij, aanpassing bovenruimte
<code>\LR</code>	last row	maak rij, aanpassing onderruimte
<code>\MR</code>	mid row	maak rij, aanpassing boven- en onderruimte
<code>\SR</code>	separate row	maak rij, aanpassing boven- en onderruimte
<code>\VL</code>	vertical line	maak een verticale lijn tot de volgende kolom
<code>\NC</code>	next column	ga naar de volgende kolom
<code>\HL</code>	horizontal line	maak een horizontale lijn
<code>\DL</code>	division line*	maak een horizontale lijn over de kolombreedte
<code>\DL[n]</code>	division line*	maak een horizontale lijn over n kolommen
<code>\DC</code>	division column*	maak een spatie over de breedte van de kolom
<code>\DR</code>	division row*	maak een rij, aanpassing onder- en bovenruimte
<code>\LOW{tekst}</code>	—	verlaag <i>tekst</i>
<code>\TWO, \THREE</code> etc.	—	gebruik ruimte over <i>twee, drie</i> kolommen

* `\DL`, `\DC` en `\DR` worden in combinatie gebruikt.

Tabel 11.4 `CONTEXT` tabel formaatcommando's.

```
[tab:effect van commandos]
{Effect van formaatcommandos.}
\startcombinatie[2*1]%
{\starttabel[|c|c|]%
\HL
\VL \bf Jaar \VL \bf Inwoners \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptabel}
{standaard}
{\starttabel[|c|c|]%
\HL
\VL \bf Jaar \VL \bf Inwoners \VL\NR
\HL
\VL 1675 \VL ~428 \VL\NR
\VL 1795 \VL 1124 \VL\NR
\VL 1880 \VL 2405 \VL\NR
\VL 1995 \VL 7408 \VL\NR
\HL
\stoptabel}
```

Tabellen

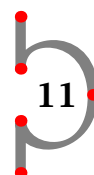
```
{alleen \type{\NR}}
\stopcombinatie
```

In het bovenstaande voorbeeld worden in de eerste tabel \SR, \FR, \MR en \LR gebruikt. Deze commando's zorgen voor een correcte interlinie. Zoals u hieronder kunt zien, zorgt \NR alleen maar voor een nieuwe rij.

Jaar	Inwoners
1675	428
1795	1124
1880	2405
1995	7408

Jaar	Inwoners
1675	428
1795	1124
1880	2405
1995	7408

standaard
alleen \NR



Tabel 11.5 Effect van formaatcommandos.

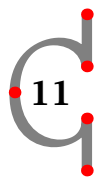
In het voorbeeld hieronder worden voorbeelden van het instellen van kolomspatiëring door middel van s0 en s1 aanduidingen getoond.

```
\plaatstabel
[hier,forceer]
[tab:inwoneraantallen]
{Effect van formaataanduidingen.}
\startcombinatie[3*2]
{\starttabel[|c|c|]
\HL
\VL \bf Jaar \VL \bf Inwoners \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptabel}
{standaard}
{\starttabel[s0 | c | c |]
\HL
\VL \bf Jaar \VL \bf Inwoners \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
```

Tabellen

```
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptabel}
{\type{s0}}
{\starttabel[| s0 c | c |]
\HL
\VL \bf Jaar \VL \bf Inwoners \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptabel}
{\type{s0} in kolom~1}
{\starttabel[| c | s0 c |]
\HL
\VL \bf Jaar \VL \bf Inwoners \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptabel}
{\type{s0} in kolom~2}
{\starttabel[s1 | c | c |]
\HL
\VL \bf Jaar \VL \bf Inwoners \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptabel}
{\type{s1}}
{}
{}
\stopcombinatie
```

Na verwerking zullen de tabellen er als volgt uitzien.



Tabellen

Jaar	Inwoners
1675	428
1795	1124
1880	2405
1995	7408

standaard

Jaar	Inwoners
1675	428
1795	1124
1880	2405
1995	7408

s0

Jaar	Inwoners
1675	428
1795	1124
1880	2405
1995	7408

s0 in kolom 1

Jaar	Inwoners
1675	428
1795	1124
1880	2405
1995	7408

s0 in kolom 2

Jaar	Inwoners
1675	428
1795	1124
1880	2405
1995	7408

s1

Tabel 11.6 Effect van formaataanduidingen.

De standaard tabel heeft een kolomspatiëring van $s2$.

Kolommen worden vaak gescheiden door een verticale lijn en rijen door middel van een horizontale lijn.

```

\plaatstabel
[hier, forceer]
[tab:divisions]
{Effect of options.}
\starttabel[|c|c|c|]
\NC Steenwijk \NC Zwartsluis \NC Hasselt \NC\S\R
\DC \DL \DC \DR
\NC Zwartsluis \VL Hasselt \VL Steenwijk \NC\S\R
\DC \DL \DC \DR
\NC Hasselt \NC Steenwijk \NC Zwartsluis \NC\S\R
\stoptabel
    
```

Steenwijk	Zwartsluis	Hasselt
Zwartsluis	Hasselt	Steenwijk
Hasselt	Steenwijk	Zwartsluis

Tabel 11.7 Effect of options.

Een zinvoller voorbeeld vindt u hieronder.

Tabellen

```

\plaatstabel
  [hier,forceer]
  [tab:voorbeeldcontextcommandos]
  {Effect van \CONTEXT||formaatcommando's.}
\starttabel[|1|c|c|c|c|]
\HL
\VL \FIVE \JustCenter{Gemeenteraadsverkiezingen 1994} \VL\SR
\HL
\VL \LOW{Partij} \VL \THREE{Districten} \VL \LOW{Totaal} \VL\SR
\DC \DL[3] \DC \DR
\VL \VL 1 \VL 2 \VL 3 \VL \VL\SR
\HL
\VL PvdA \VL 351 \VL 433 \VL 459 \VL 1243 \VL\FR
\VL CDA \VL 346 \VL 350 \VL 285 \VL ~981 \VL\MR
\VL VVD \VL 140 \VL 113 \VL 132 \VL ~385 \VL\MR
\VL HKV/RPF/SGP \VL 348 \VL 261 \VL 158 \VL ~767 \VL\MR
\VL GPV \VL 117 \VL 192 \VL 291 \VL ~600 \VL\LR
\HL
\stoptabel
  
```

In de laatste kolom wordt een ~ gebruikt om een viercijferig getal te simuleren. Het ~ heeft de breedte van een cijfer.

Gemeenteraadsverkiezingen 1994				
Partij	Districten			Totaal
	1	2	3	
PvdA	351	433	459	1243
CDA	346	350	285	981
VVD	140	113	132	385
HKV/RPF/SGP	348	261	158	767
GPV	117	192	291	600

Tabel 11.8 Effect van CONTEXT-formaatcommando's.

Soms worden tabellen te groot en is het wenselijk het korps of de rij- en kolomspatiëring aan te passen. Dergelijke aanpassingen worden gedaan met:

```
\steltabellenin[...=...]
```

Tabellen

```

\plaatstabel
[hier,forceer]
[tab:steltabellenin]
{Gebruik van \type{\steltabellenin}.}
\startcombinatie[1*3]
{\steltabellenin[korps=10pt]
\starttabel[|c|c|c|c|c|c|]
\HL
\VL \use6 \JustCenter{Afname van rijkdom in guldens (Df1)} \VL\SR
\HL
\VL Year \VL 1.000--2.000
\VL 2.000--3.000
\VL 3.000--5.000
\VL 5.000--10.000
\VL over 10.000 \VL\SR

\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR
\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptabel}
{\tt korps=10pt}
{\steltabellenin[korps=8pt]
\starttabel[|c|c|c|c|c|c|]
\HL
\VL \use6 \JustCenter{Afname van rijkdom in guldens (Df1)} \VL\SR
\HL
\VL Year \VL 1.000--2.000
\VL 2.000--3.000
\VL 3.000--5.000
\VL 5.000--10.000
\VL over 10.000 \VL\SR

\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR
\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptabel}
{\tt korps=8pt}

```

Tabellen

```
{\steltabellenin[korps=6pt, afstand=klein]
\starttabel[|c|c|c|c|c|c|]
\HL
\VL \use6 \JustCenter{Afname van rijkdom in guldens (Df1)} \VL\SR
\HL
\VL Year \VL 1.000--2.000
        \VL 2.000--3.000
        \VL 3.000--5.000
        \VL 5.000--10.000
        \VL over 10.000 \VL\SR
\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR
\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptabel}
{\tt korps=6pt, afstand=klein}
\stopcombinatie
```

U kunt tevens de tabel als geheel instellen met:

```
\stelplaatsblokkenin[...=...]
```

De nummering en de labels worden ingesteld met:

```
\stelblokkopjesin[...=...]
```

De genoemde commando's worden in het instelgebied van de invoerfile geplaatst.

```
\stelplaatsblokkenin[plaats=links]
\stelblokkopjesin[letter=vetschuin]
\plaatstabel{Openingstijden bibliotheek.}
\starttabel[|c|c|]
\HL
\VL \bf Dag \VL \use2 \bf Openingstijden \VL\SR
\HL
\VL Maandag \VL 14.00 -- 17.30 \VL 18.30 -- 20.30 \VL\FR
\VL Dinsdag \VL \VL \VL\MR
\VL Woensdag \VL 10.00 -- 12.00 \VL 14.00 -- 17.30 \VL\MR
```

Paragraaf opmaak

Afname van rijkdom in guldens (Dfl)					
Year	1.000-2.000	2.000-3.000	3.000-5.000	5.000-10.000	over 10.000
1675	22	7	5	4	5
1724	4	4	-	4	3
1750	12	3	2	2	-
1808	9	2	-	-	-

korps=10pt

Afname van rijkdom in guldens (Dfl)					
Year	1.000-2.000	2.000-3.000	3.000-5.000	5.000-10.000	over 10.000
1675	22	7	5	4	5
1724	4	4	-	4	3
1750	12	3	2	2	-
1808	9	2	-	-	-

korps=8pt

Afname van rijkdom in guldens (Dfl)					
Year	1.000-2.000	2.000-3.000	3.000-5.000	5.000-10.000	over 10.000
1675	22	7	5	4	5
1724	4	4	-	4	3
1750	12	3	2	2	-
1808	9	2	-	-	-

korps=6pt, afstand=klein

Tabel 11.9 Gebruik van `\steltabellenin`.

```

\VL Donderdag \VL 14.00 -- 17.30 \VL 18.30 -- 20.30 \VL\MR
\VL Vrijdag \VL 14.00 -- 17.30 \VL \VL\MR
\VL Zaterdag \VL 10.00 -- 12.30 \VL \VL\LR
\HL
\stoptabel
    
```



In sommige situaties is het noodzakelijk om informatie in een dusdanige vorm te geven dat deze overzichtelijker wordt weergegeven.

Dag	Openingstijden	
Maandag	14.00 - 17.30	18.30 - 20.30
Dinsdag		
Woensdag	10.00 - 12.00	14.00 - 17.30
Donderdag	14.00 - 17.30	18.30 - 20.30
Vrijdag	14.00 - 17.30	
Zaterdag	10.00 - 12.30	

Tabel 11.10 *Openingstijden bibliotheek.*

12

```
\definieeralineas[...][...=...]
```

```
\stelalineasin[.1.][.2.][...=...]
```

In het onderstaande voorbeeld wordt getoond hoe alineas die uit meerdere regels bestaan, naast elkaar kunnen worden geplaatst.

```
\definieeralineas
  [mijnpar][n=3,voor={\blanko},na={\blanko}]

\stelalineasin
  [mijnpar][1][breedte=.1\tekstbreedte,letter=vet]
\stelalineasin
  [mijnpar][2][breedte=.4\tekstbreedte]

\startmijnpar
1252
\mijnpar
Hasselt verkrijgt stadsrechten van bisschop Hendrik van
Vianden.
\mijnpar
Hendrik van Vianden werd door andere steden onder druk gezet
om de stadsrechten niet te verlenen. Het kostte Hasselt veel
tijd om de bisschop te overtuigen. Nadat Hasselt de bisschop
had gesteund bij een kleine oorlog tegen de Drenten werden
de stadsrechten verleend.
\stopmijnpar

\startmijnpar
```

1350

`\mijnpar`

Hasselt treedt toe tot het Hanzepact om haar internationale handel te beschermen.

`\mijnpar`

Het Hanzepact was van groot belang voor de handelaren van Hasselt. Goederen werden in die dagen door iedere stad, op iedere hoofdstraat en bij iedere rivieroversteekplaats apart belast. Belastingvrije routes door heel Europa was een van de voordelen van het lidmaatschap. Hasselt is altijd een van de kleinere leden van het pact geweest.

`\stopmijnpar`

Na het definiëren van een alinea met `\definieeralineas` kan de layout van de alinea met `\stelalineasin` worden vastgelegd. Vervolgens kan iedere alinea worden begonnen met `\start...` en `\stop...`. Een nieuwe alinea begint telkens met de naam van de gedefinieerde alinea, in dit geval `\mijnpar`.

- | | | |
|------|---|---|
| 1252 | Hasselt verkrijgt stadsrechten van bisschop Hendrik van Vianden. | Hendrik van Vianden werd door andere steden onder druk gezet om de stadsrechten niet te verlenen. Het kostte Hasselt veel tijd om de bisschop te overtuigen. Nadat Hasselt de bisschop had gesteund bij een kleine oorlog tegen de Drenten werden de stadsrechten verleend. |
| 1350 | Hasselt treedt toe tot het Hanzepact om haar internationale handel te beschermen. | Het Hanzepact was van groot belang voor de handelaren van Hasselt. Goederen werden in die dagen door iedere stad, op iedere hoofdstraat en bij iedere rivieroversteekplaats apart belast. Belastingvrije routes door heel Europa was een van de voordelen van het lidmaatschap. Hasselt is altijd een van de kleinere leden van het pact geweest. |

Een tweede voorbeeld van het herinrichten van alineas vindt u hieronder.

```
\definieeralineas[chempar][n=3,voor=,na=,afstand=1em]
```

```
\stelalineasin[chempar][1][breedte=.12\tekstbreedte]
```

```
\stelalineasin[chempar][2][breedte=.4\tekstbreedte]
```

```
\startchempar
```

Kalkoven

Paragraaf opmaak

```
\chempar
```

Hasselt heeft zijn eigen kalkovens. Deze werden in 1504 gebouwd en produceerden kalk tot 1956. Tegenwoordig vormen de ovens een toeristische attractie.

```
\chempar
```

```
\chemie{CaCO_3,~,GIVES,~,CaO,~,+,~,CO_2}
```

```
\stopchempar
```

Dit komt er ongeveer als volgt uit te zien:

Kalkoven Hasselt heeft zijn eigen kalkovens. $\text{CaCO}_3 \rightarrow \text{CaO} + \text{CO}_2$
Deze werden in 1504 gebouwd en produceerden kalk tot 1956. Tegenwoordig vormen de ovens een toeristische attractie.

Er is tevens een verkorte schrijfwijze beschikbaar:

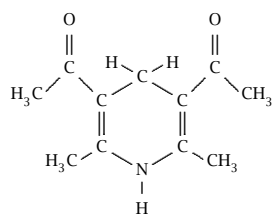
```
\chempar Kalkoven
```

```
\\ Hasselt heeft zijn eigen kalkovens. Deze werden in  
1504 gebouwd en produceerden kalk tot 1956.  
Tegenwoordig vormen de ovens een touristische  
attractie.
```

```
\\ \chemie{CaCO_3,~,GIVES,~,CaO,~,+,~,CO_2} \\
```

De `\\` worden als kolomscheider gebruikt en zijn essentieel.

De chemische module wordt toegelicht in een andere handleiding, omdat lang niet iedereen in chemie is geïnteresseerd. Chemische structuren zien er overigens wel heel indrukwekkend uit.



Verbinding A

CONTEXT maakt gebruik van de PICTEX macros van M.J. Wichura om deze structuren te tekenen. Hoewel de chemische module slechts uit twee commando's bestaat, is er wel enige oefening nodig om de juiste resultaten te verkrijgen. De invoer voor de formule ziet er als volgt uit:

```
\plaatsformule[-]
```

```
\startformule
```

Kolommen

```
\startchemie[schaal=klein,breedte=passend,boven=3000,onder=3000]
  \chemie[SIX,SB2356,DB14,Z2346,SR3,RZ3,-SR6,+SR6,-RZ6,+RZ6]
    [C,N,C,C,H,H,H]
  \chemie[PB:Z1,ONE,ZO,DIR8,ZO,SB24,DB7,Z27,PE][C,C,CH_3,0]
  \chemie[PB:Z5,ONE,ZO,DIR6,ZO,SB24,DB7,Z47,PE][C,C,H_3C,0]
  \chemie[SR24,RZ24][CH_3,H_3C]
  \bottext{Verbinding A}
\stopchemie
\stopformule
```



Eenvoudige recht-toe recht-aan tekst kan in kolommen worden gezet. Als u het tekstfragment vooraf laat gaan door `\startkolommen` en afsluit met `\stopkolommen` wordt alles er tussenin in het opgegeven aantal kolommen gezet.

```
\startkolommen[...,.=...,.] ... \stopkolommen
```

```
\startkolommen[n=3,tolerantie=zeersoepel]
\switchnaarkorps[9pt]
```

Hasselt is een oude Hanzestad ongeveer 12 km ten noorden van Zwolle aan de rivier het Zwartewater.

De stad heeft een lange historie sinds het verkrijgen van de stadrechten in 1252. Herinneringen aan die rijke historie zijn terug te vinden in de monumenten die nog steeds het gezicht van het centrum van Hasselt bepalen.

De Sint Stephanuskerk domineert het aanzien van Hasselt. Het is een laat|gothische kerk die dateert uit 1479 met een magnifiek orgel. Het voormalige stadhuis staat op de Markt. Gebouwd tussen 1500 en 1550 herbergt het een grote collectie wapens, waaronder de grootste collectie haakbussen in de wereld.

Verder is er een korenmolen 'De Zwaluw' te vinden die

dateert uit 1748 en de 'Stenendijk', een unieke gemetselde dijk. Ook kunt u de nog steeds werkende kalkovens bezichtigen.

Het stadscentrum met de vestingwerken, het Van Stolkspark en de druk bezochte kades nodigen uit voor een wandeling.

Het gebied rond Hasselt moet ook worden genoemd. In de winter huizen in de nabijgelegen polder Mastenbroek duizenden ganzen. In de zomer groeit aan de oevers van het Zwartewater de zeldzame Kievitsbloem. Als u die gaat bekijken, komt u langs plaatsjes als Genne, Streukel en Cellemuiden. Ideaal voor een wandel- of fietstocht.

Hasselt is ook een belangrijk watersportcentrum. De meren van noord||west Overijssel, de IJssel, de Overijsselse Vecht en de randmeren zijn goed bereikbaar vanuit de jachthaven 'De Molenwaard'. Zeilen, vissen, zwemmen en kanoën zijn sporten die in Hasselt goed kunnen worden beoefend.

Jaarlijks keren bepaalde festiviteiten terug. Zo is er aan het eind van de maand augustus het Eui||festival.

\stopkolommen

Het resultaat is een drie-koloms tekst. Ten behoeve van het uitlijnen wordt het afbreek-mechanisme ingesteld met tolerantie=zeersoepel.

Hasselt is een oude Hanzestad ongeveer 12 km ten noorden van Zwolle aan de rivier het Zwartewater.

De stad heeft een lange historie sinds het verkrijgen van de stadrechten in 1252. Herinneringen aan die rijke historie zijn terug te vinden in de monumenten die nog steeds het gezicht van het centrum van Hasselt bepalen.

De Sint Stephanuskerk domineert het aanzien van Hasselt. Het is een laat-gothische kerk die dateert uit 1479 met een magnifiek orgel. Het voormalige stadhuis staat op de

Markt. Gebouwd tussen 1500 en 1550 herbergt het een grote collectie wapens, waaronder de grootste collectie haakbussen in de wereld.

Verder is er een korenmolen 'De Zwaluw' te vinden die dateert uit 1748 en de 'Stenendijk', een unieke gemetselde dijk. Ook kunt u de nog steeds werkende kalkovens bezichtigen.

Het stadscentrum met de vestingwerken, het Van Stolkspark en de druk bezochte kades nodigen uit voor een wandeling.

Het gebied rond Hasselt moet ook worden genoemd. In de winter huizen in de nabijgelegen polder Mastenbroek duizenden ganzen. In de zomer groeit aan de oevers van het Zwartewater de zeldzame Kievitsbloem. Als u die gaat bekijken, komt u langs plaatsjes als Genne, Streukel en Cellemuiden. Ideaal voor een wandel- of fietstocht.

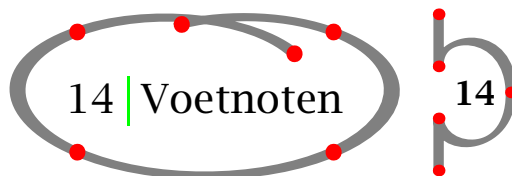
Hasselt is ook een belangrijk watersportcentrum. De meren van noord-west Overijssel, de IJssel, de Overijsselse Vecht en de randmeren zijn goed bereikbaar vanuit de jachthaven 'De Molenwaard'. Zeilen, vis-

Voetnoten

sen, zwemmen en kanoën zijn Jaarlijks keren bepaalde festi- het Eui-festival.
sporten die in Hasselt goed viteiten terug. Zo is er aan het
kunnen worden beoefend. eind van de maand augustus

Indien mogelijk kan een kolom worden afgedwongen met `\kolom`. Het instellen van kolommen gebeurt met:

```
\stelkolommenin[...]=...]
```



Indien u een tekst wilt annoteren met voetnoten gebruikt u `\voetnoot`. Het commando ziet er als volgt uit:

```
\voetnoot[ref]{...}
```

De vierkante haken zijn optioneel en bedoeld voor een logische naam waarmee het voetnootnummer meerdere malen kan worden opgeroepen. De accolades bevatten de tekst die in de voetnoot moet worden weergegeven.

Een voetnootnummer kan worden opgeroepen met:

```
\noot[ref]
```

Indien u de onderstaande tekst heeft ingevoerd:

De Hanze was een laat||middeleeuws commercieel verbond van steden in het Noorden en rond de Baltische Zee. Het verbond werd gesloten voor het bevorderen en de bescherming van de handel van haar leden.\voetnoot[oorlog]{Dit was de bron van veel jaloezie tussen steden onderling en veroorzaakte menige oorlog.} In de Hanzeperiode werd levendig gehandeld in hout, wol, metaal, stoffen, zout, wijn en bier. De handel

veroorzaakte grote groei in de Hanzesteden\voetnoot{Hasselt was ook zo'n stad.} en het lokte vele oorlogen uit.\noot[oorlog]

Krijgt u:

De Hanze was een laat-middeleeuws commercieel verbond van steden in het Noorden en rond de Baltische Zee. Het verbond werd gesloten voor het bevorderen en de bescherming van de handel van haar leden.⁵ In de Hanzeperiode werd levendig gehandeld in hout, wol, metaal, stoffen, zout, wijn en bier. De handel veroorzaakte grote groei in de Hanzesteden⁶ en het lokte vele oorlogen uit.⁵

Het nummeren van voetnoten gebeurt automatisch. De wijze van weergave van voetnoten wordt ingesteld met het commando:

15

```
\stelvoetnotenin[...]=...]
```

15 | Definities

Om definities, begrippen en concepten enigszins consistent weer te geven, wordt gebruik gemaakt van:

```
\doordefinieren[...][...]=...]
```

Bijvoorbeeld:

```
\doordefinieren  
  [concept]  
  [plaats=aanelkaar,kopletter=vet,breedte=ruim]  
  
  \concept{Hasselter juffer} Een traditioneel koekje van  
  besuikerd bladerdeeg. Erg zoet en erg lekker. Vergelijkbaar  
  met het Arnhemse meisje (krakeling). \par
```

⁵ Dit was de bron van veel jaloezie tussen steden onderling en veroorzaakte menige oorlog.

⁶ Hasselt was ook zo'n stad.

Een dergelijke definitie wordt als volgt weergegeven:

Hasselter juffer Een traditioneel koekje van besuikerd bladerdeeg. Erg zoet en erg lekker. Vergelijkbaar met het Arnhemse meisje (krakeling).

Maar het is ook mogelijk om een andere vormgeving te kiezen:

Hasselter bitter

Een sterk alcoholische drank (tot 40%) gemengd met kruiden om een specifieke smaak te verkrijgen. Het wordt verkocht in een stenen kruik en dient koud te worden geserveerd.

Euifeest Een hooifeest om het einde van een periode van hard werken af te sluiten. De festiviteiten vinden in de derde week van augustus plaats.

Indien de definitie uit meer dan een alinea bestaat kunt u `\start... \stop...` gebruiken.

```
\doordefinieren
 [concept]
 [plaats=rechts,
 kopletter=vet,
 breedte=ruim]
```

```
\startconcept{Euifeest}
 Een hooifeest om een periode van hard werken af te sluiten.
 Het feest vindt plaats aan het eind van augustus en duurt
 een week.
```

```
 De stad wordt volledig versierd en verlicht en dit heeft
 Hasselt de bijnaam {\em Lichtstad van het Noorden} gegeven.
 Het feest eindigt in een traditionele braderie.
```

```
\stopconcept
```

Dit wordt:

Een hooifeest om een periode van hard werken af te sluiten. Het feest vindt plaats aan het eind van augustus en duurt een week. **Euifeest**

De stad wordt volledig versierd en verlicht en dit heeft Hasselt de bijnaam *Lichtstad van het Noorden* gegeven. Het feest eindigt in een traditionele braderie.

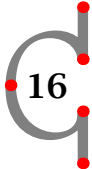
De vormgeving kan worden vastgelegd binnen het tweede paar van vierkante haken in `\doordefinieren[...]`. Meestal wordt dit echter gedaan met het commando:

```
\steldoordefinierenin[...][...=...]
```


16 | Genummerde definities

Met `\doornummeren` worden tekstelementen als opmerkingen en vragen (antwoorden) gedefinieerd. Genummerde opmerkingen worden verkregen door:

```
\doornummeren[...1...][.2.][...=...]
```



Bijvoorbeeld:

```
\doornummeren
  [opmerking]
  [plaats=boven,
   tekst=Opmerking,
   tussen=\blanko,
   na=\blanko]
```

Na definitie zijn de commando's `\opmerking`, `\subopmerking`, `\resetopmerking` en `\volgendeopmerking` beschikbaar. Opmerkingen worden nu als volgt ingevoerd:

`\opmerking` Aan het begin van de middeleeuwen was Hasselt een bedevaartsoord. De zogenaamde `{\em Heilige Stede}` werd afgebroken gedurende de reformatie. Na 300 jaar werd in 1930 de `{\em Heilige Stede}` weer heropend.

`\subopmerking` Tegenwoordig is de `{\em Heilige Stede}` gesloten en is slechts eens per jaar geopend tijdens een kerkdienst in de open lucht.

Opmerking 1

Aan het begin van de middeleeuwen was Hasselt een bedevaartsoord. De zogenaamde *Heilige Stede* werd afgebroken gedurende de reformatie. Na 300 jaar werd in 1930 de *Heilige Stede* weer heropend.

Opmerking 1.1

Tegenwoordig is de *Heilige Stede* gesloten en is slechts eens per jaar geopend tijdens een kerkdienst in de open lucht.

Genummerde definities

U kunt het nummeren herstellen `\resetopmerking` of `\resetsubopmerking` of ophogen met `\volgendeopmerking` of `\volgendesubopmerking`. Normaal gebeurt dit automatisch per hoofdstuk.

De layout van genummerde definities wordt ingesteld met `\doornummeren` of met:

```
\steldoornummerenin[...][...=...]
```

De layout van `\opmerking` en `\subopmerking` uit het voorbeeld kan worden ingesteld met:

```
\steldoornummerenin[opmerking][kopletter=vet]
\steldoornummerenin[subopmerking][kopletter=schuin]
```

Indien een nummer niet nodig is, typt u:

```
\opmerking[-]
```

In het geval dat een tekstelement uit meerdere alineas bestaat, moet het commando-paar `\startopmerking` ... `\stopopmerking` worden gebruikt. Dit commando-paar is beschikbaar na definitie van **Opmerking** met `\doornummeren[opmerking]`.

Een langere definitie zou er als volgt uit kunnen zien:

```
\doordefinieren
[concept]
[plaats=rechts,
kopletter=vet,
breedte=ruim]
```

```
\startconcept{Euifeest}
```

Een hooifeest om een periode van hard werken af te sluiten. Het feest vindt plaats aan het eind van augustus en duurt een week.

De stad wordt volledig versierd en verlicht en dit heeft Hasselt de bijnaam `\em Lichtstad van het Noorden` gegeven. Het feest eindigt in een traditionele braderie.

```
\stopconcept
```

Opmerking 1

Aan het begin van de middeleeuwen was Hasselt een bedevaartsoord. De zogenaamde *Heilige Stede* werd afgebroken gedurende de reformatie. Na 300 jaar werd in 1930 de *Heilige Stede* weer heropend.

Tegenwoordig is de *Heilige Stede* gesloten en slechts eens per jaar geopend tijdens een kerkdienst in de open lucht.



Tekst kan met `\omlijnd` worden `omlijnd`. Het commando ziet er als volgt uit:

```
\omlijnd[...=...]{...}
```

De vierkante haken zijn optioneel en bevatten de instellingen. De accolades bevatten de te omlijnen tekst.

```
\omlijnd[hoogte=3em,breedte=passend]{Dit is een grote box}
```

Dit wordt:

Dit is een grote box

Een ander voorbeeld van `\omlijnd` met de instellingen wordt hieronder getoond.

```
\stelkleurenin  
[status=start]  
\regellinks  
{\omlijnd[breedte=passend]{Inwoners van Hasselt}}  
\regelmidden  
{\omlijnd[hoogte=1.5cm,kader=uit]{hebben een}}  
\regelrechts  
{\omlijnd[achtergrond=raster]{historische achtergrond}}
```

Dit leidt tot:

Inwoners van Hasselt

hebben een

historische achtergrond

Het `\omlijnd`-commando is zeer geavanceerd en wordt in zeer veel macro's ingezet. Het commando wordt ingesteld met:

```
\stelomlijndin[...]=...]
```

18 | Omlijnde paragrafen

Voor het omkaderen van complete alinea's wordt het volgende commando-paar gebruikt:

```
\startkadertekst[...][...]=...] ... \stopkadertekst
```

```
\definieerplaatsblok[intermezzo]
```

```
\plaatsintermezzo[hier][blok;brug]{Een intermezzo.}
```

```
\startkadertekst[breedte=.8\zetbreedte]
```

Een brug over het Zwartewater was essentieel voor Hasselt. De bisschop van Utrecht gaf zijn toestemming voor de bouw in 1486.

```
\blanko
```

Andere steden in de omgeving van Hasselt waren bang voor de hoge tolgelden die Hasselt bij passage over zo'n brug zou kunnen vragen. Deze steden hebben de bouw lange tijd tegengehouden.

```
\stopkadertekst
```

Dit voorbeeld illustreert het commando `\definieerplaatsblok`. Meer informatie vindt u in paragraaf 37.2. De `\blanko` is noodzakelijk om een lege regel af te dwingen.

Een brug over het Zwartewater was essentieel voor Hasselt. De bisschop van Utrecht gaf zijn toestemming voor de bouw in 1486.

Andere steden in de omgeving van Hasselt waren bang voor de hoge tolgelden die Hasselt bij passage over zo'n brug zou kunnen vragen. Deze steden hebben de bouw lange tijd tegengehouden.

Intermezzo 18.1 Een intermezzo.

Het omkaderen wordt ingesteld met:

```
\stelkaderetekstenin[...]=...]
```



19

Het is eenvoudig om teksten in de marge te plaatsen. Het commando daarvoor is `\inmarge`.

```
\inmarge[.1.][ref]{.2.}
```

In een van de eerdere voorbeelden is al met `\inmarge` gewerkt.

```
\inmarge  
{\externfiguur  
 [hass23g]  
 [type=tif,  
  breedte=\margebreedte,  
  kader=aan]}
```



Dit resulteert in een figuur in de marge. Een figuur in de marge is natuurlijk erg smal en te klein om goed te kunnen weergeven.

Hieronder worden enkele voorbeelden van margeteksten gegeven.

De Ridderstraat `\inmarge{Ridderstraat}` heeft een logische naam. In de 14e en 15e eeuw woonde de adel en prominente mensen in deze straat. Enkele van hun grote huizen staan er nog en zijn later in gebruik genomen als armenhuis `\inrechter{armen-\\huis}` en bejaardenhuis.

Tot aan `\inlinker[laag]{\tfd 1940}`1940 was er een synagoge in de Ridderstraat. Ongeveer 40 joodse inwoners van Hasselt vierden daar hun Sabbat. Tijdens de oorlog werden deze joden naar Westerbork gestuurd en vandaar uit verder

Pagina's en paginanummering

getransporteerd naar de vernietigingskampen in Duitsland en Polen. Geen van de joodse families keerden terug. De synagoge werd in 1958 afgebroken.

De commando's `\inmarge`, `\inlinker` en `\inrechter` hebben allen een vergelijkbare functie. In een tweezijdig document plaatst `\inmarge` de margeteksten automatisch in de correcte marge. De `\` zijn om regelafbreking af te dwingen. Het voorbeeld komt er als volgt uit te zien:

Ridder- straat

De Ridderstraat heeft een logische naam. In de 14e en 15e eeuw woonde de adel en prominente mensen in deze straat. Enkele van hun grote huizen staan er nog en zijn later in gebruik genomen als armenhuis en bejaardenhuis.

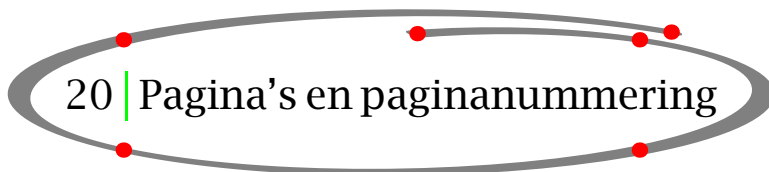
1940

Tot aan 1940 was er een synagoge in de Ridderstraat. Ongeveer 40 joodse inwoners van Hasselt vierden daar hun Sabbat. Tijdens de oorlog werden deze joden naar Westerbork gestuurd en vandaar uit verder getransporteerd naar de vernietigingskampen in Duitsland en Polen. Geen van de joodse families keerden terug. De synagoge werd in 1958 afgebroken.

Margeteksten worden ingesteld met:

```
\stelinmargesin
```

armen-
huis



Een pagina-overgang wordt afgedwongen of geblokkeerd met:

```
\pagina[...]
```

De eventuele opties worden tussen vierkante haakjes geplaatst. De opties worden in tabel 20.1 toegelicht.

Paginanummering vindt automatisch plaats, maar nummers kunnen worden afgedwongen met:

```
\pagina[25]
```

Soms is het beter om een relatief paginanummer in te voeren (indien vooraf niet bekend is welk paginanummer de laatste pagina heeft) [+2] of [-2].



Optie	Betekenis
ja	gedwongen pagina-overgang
opmaak	gedwongen pagina-overgang zonder opvullen
nee	geen pagina-overgang
voorkeur	bij voorkeur hier een pagina-overgang
grotevoorkeur	bij grote voorkeur hier een pagina-overgang
links	volgende pagina is een linker pagina
rechts	volgende pagina is een rechter pagina
blokkeer	volgende commando geen effect
reset	volgende commando heeft effect
leeg	voeg een lege pagina toe
laatste	vul pagina's aan tot een even nummer
viertal	vul pagina's aan tot een viertal is bereikt

Tabel 20.1 Pagina opties.

De positie van de paginanummers op een pagina hangen af van uw eigen voorkeur en of het gaat om een enkel- of dubbelzijdig document. Het nummeren van pagina's wordt ingesteld met:

```
\stelnummeringin[...]=...]
```

De voorkeuren worden tussen de vierkante haakjes geplaatst.

Tabellen en figuren nemen nogal wat ruimte in op een pagina. Het plaatsen van dergelijke tekstelementen kan tijdelijk worden uitgesteld, zodat u zelf kunt bepalen waar de figuren en tabellen worden geplaatst. Dit wordt gedaan met:

```
\startuitstellen ... \stopuitstellen
```

Als u bijvoorbeeld typt:

```
\startuitstellen
\plaatsfiguur
{Een uitgestelde figuur.}
{\externfiguur[hass16g][kader=aan,breedte=\tekstbreedte]}
\pagina
\stopuitstellen
```

De figuur wordt bovenaan de volgende pagina geplaatst, waardoor de lopende tekst zo min mogelijk wordt verstoord.



Documenten hebben soms hoofd- en voetteksten die voor allerlei doeleinden worden gebruikt. De commando's om hoofd- en voetteksten te plaatsen zijn:

```
\stelvoettekstenin[.1.][.2.][.3.]
```

```
\stelhoofdkostenin[.1.][.2.][.3.]
```



Het eerste paar haken is optioneel en bedoeld voor de locatie van de voet- of hoofdtekst (tekst, rand enz.). De voet- en hoofdtekst zelf worden tussen de overige vierkante haken geplaatst. In een enkelzijdig document zijn alleen het tweede en derde paar vierkante haken nodig. Het tweede paar van bijvoorbeeld `\stelvoettekstenin` bevat de tekst linksonder en het tweede paar de tekst die rechtsonder moet komen te staan. In een dubbelzijdig document zijn nog twee paren beschikbaar voor teksten in de voet van de linker pagina.

```
\stelvoettekstenin[Handleiding][paragraaf]
```

In dit voorbeeld verschijnt de tekst *Handleiding* in de linker onderhoek van de pagina en titel van de actuele paragraaf in de rechteronderhoek. Deze voettekst verandert automatisch bij overgang naar een volgende paragraaf.

De hoofd- en voetteksten kunnen worden ingesteld met de commando's:

```
\stelhoofdin[...][...,...=...,...]
```




Figuur 20.1 Een uitgestelde figuur.

```
\stelvoetin[...][...,...=...,...]
```

Als de hoofd- en/of voetteksten op een bepaalde pagina niet nodig zijn, typt u:

```
\geenhoofdenvoetregels
```

22 | Inhoudsopgave (lijsten)

Een inhoudsopgave bevat hoofdstuktitels, hoofdstuknummers en paginanummers en kan worden uitgebreid met de paragrafen, subparagrafen enz. Een inhoudsopgave wordt automatisch gegenereerd door in te typen:

```
\plaatsinhoud
```

Afhankelijk van de locatie in het document wordt vervolgens een inhoudsopgave aangemaakt. Aan het begin van een document wordt een volledige inhoudsopgave gegenereerd, bestaande uit hoofdstukken, paragrafen, subparagrafen enz. Aan het begin van een hoofdstuk echter:

```
\hoofdstuk{Hasselt in de zomer}
```

```
\plaatsinhoud
```

```
\paragraaf{Hasselt in juli}
```

```
\paragraaf{Hasselt in augustus}
```

wordt alleen een lijst van (sub)paragraaftitels en de corresponderende paragraafnummers en paginanummers geplaatst.

Het commando `\plaatsinhoud` is beschikbaar nadat een zogenaamde samengestelde lijst is gedefinieerd met:

```
\definieersamengesteldelijst[.1.][..., .2.,...][...,...=...,...]
```

Dit commando en `\definieerlijst` staan u toe om uw eigen lijsten aan te maken die nodig zijn om een document te structureren.

Het gebruik van deze commando's wordt geïllustreerd aan de inhoudsopgave.

Inhoudsopgave (lijsten)

```
\definieerlijst[hoofdstuk]
```

```
\stellijstin
```

```
[hoofdstuk]
```

```
[voor=\blanko,
```

```
na=\blanko,
```

```
letter=vet]
```

```
\definieerlijst[paragraaf]
```

```
\stellijstin
```

```
[paragraaf]
```

```
[variant=d]
```

Tijdens het verwerken van de invoerfile worden twee lijsten aangemaakt die gecombineerd worden tot één inhoudsopgave met `\defineersamengesteldelijst`.

```
\defineersamengesteldelijstin
```

```
[inhoud]
```

```
[hoofdstuk,paragraaf]
```

```
[niveau=subparagraaf]
```

22

Er zijn nu twee commando's beschikbaar `\plaatsinhoud` en `\volledigeinhoud`. Met het tweede commando wordt tevens de titel *inhoud* boven de inhoudsopgave geplaatst.

De vormgeving van de lijsten kan worden beïnvloed met de parameter `variant`.

Alternatief	Weergave
a	nummer - titel - paginanummer
b	nummer - titel - spaties - paginanummer
c	nummer - titel - dots - paginanummer
d	nummer - titel - paginanummer (continu)
e	gereserveerd voor interactieve documenten
f	gereserveerd voor interactieve documenten

Tabel 22.1 Alternatieven voor lijstweergave.

De lijsten worden ingesteld met:

```
\stellijstin[.,.,.,.,.][.,.,.=.,.,.]
```

Inhoudsopgave (lijsten)

```
\stelsamengesteldelijstin[...] [,,,=,,,]
```

Indien u de layout van een inhoudsopgave wilt veranderen dan moet u in gedachten houden dat het om een lijst gaat.

```
\stelsamengesteldelijstin  
[inhoud]  
[variant=c,  
titeluitlijnen=nee,  
breedte=2.5cm]
```

Het resultaat is een iets andere layout dan de standaard layout.

Lijsten worden geplaatst met:

```
\plaatslijst[,,,,,] [,,,=,,,]
```

Indien u een inhoud plaatst, kunt u bijvoorbeeld intypen:

```
\plaatslijst[inhoud] [niveau=paragraaf]
```

of:

```
\plaatsinhoud[niveau=paragraaf]
```

Dan worden alleen de paragrafen in de inhoudsopgave opgenomen. Een dergelijke optie komt van pas bij documenten waarin bijvoorbeeld subsubsubsubsubparagrafen voorkomen die u niet in de inhoudsopgave wilt opnemen.

Een lange lijst of een grote inhoudsopgave neemt meer dan één pagina in beslag. Om een pagina-overgang af te kunnen dwingen is het volgende commando beschikbaar:

```
\volledigeinhoud[2.2,8.5,12.3.3]
```

Een nieuwe pagina wordt gegenereerd na paragraaf 2.2 en 8.5 en subparagraaf 12.3.3.

Sporadisch heeft u wellicht de behoefte om teksten in uw inhoudsopgave tussen te voegen. Dit wordt gedaan met:

```
\schrijfnaarlijst[.1.]{.2.}{.3.}
```

```
\schrijftussenlijst[.1.]{.2.}
```

Indien u bijvoorbeeld een opmerking in uw inhoudsopgave wilt maken vlak na een paragraaf met de titel *Hotels in Hasselt* dan kunt u bijvoorbeeld intypen:

```
\paragraaf{Hotels in Hasselt}
\schrijfn naarlijst[paragraaf]{\blanko}
\schrijftussenlijst[paragraaf]{---moet nog worden aangemaakt---}
\schrijfn naarlijst[paragraaf]{\blanko}
```

23 | Registers

23

Het is mogelijk om een of meer registers te genereren. Standaard is hiervoor het commando `\index` beschikbaar. Voor het toevoegen van een woord aan de lijst typt u bijvoorbeeld:

```
\index{stadhuis}
```

Het woord *stadhuis* zal als een ingang in het register verschijnen. Een register wordt alfabetisch gesorteerd met behulp van het programma `TEXUTIL`. In sommige gevallen kan een indexwoord niet worden gesorteerd. Dit is het geval bij bepaalde symbolen. Dergelijke ingangen worden als volgt gedefinieerd:

```
\index[minteken]{$-$}
```

De ingang wordt nu gesorteerd op het woord tussen vierkante haken *minteken*.

Soms zijn er sub- en subsubingangen. Deze worden op een vergelijkbare wijze gedefinieerd:

```
\index{stadhuis+locatie}
\index{stadhuis+architectuur}
```

Het register zelf wordt (meestal aan het eind van een document) gegenereerd met:

```
\plaatsindex
```

of:

```
\volledigeindex
```

Het commando `\index` is een voorgedefinieerd `CONTEXT`-commando, maar het is ook mogelijk uw eigen registers te definiëren.

```
\definieerregister[.1.][.2.]
```

Een register gebaseerd op de straten van Hasselt kan als volgt worden gedefinieerd.

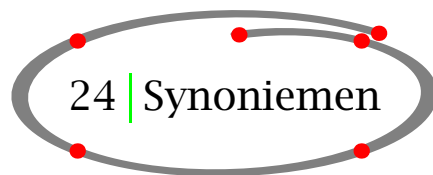
```
\definieerregister[straat][straten]
```

Op dit moment is er een registercommando `\straat` beschikbaar. Een ingang wordt dan ingevoerd met `\straat{Ridderstraat}`. Het register zelf wordt opgeroepen met:

```
\plaatsstraat  
\volledigestraat  
\plaatsregister[straat]
```

De weergave van de registers wordt ingesteld met:

```
\stelregisterin[...][.,,..=.,...]
```



In een document dienen sommige woorden consistent te worden weergegeven. Denk daarbij bijvoorbeeld aan afkortingen. Om die consistentie af te dwingen is het onderstaande commando beschikbaar:

```
\definieersynoniemen[.1.][.2.][.3.][.4.]
```

Het eerste paar haken bevat de naam van het synoniemcommando in enkelvoud, het tweede paar in meervoud. Het derde paar haken bevat een commando.

Het commando `\afkortingen` wordt gedefinieerd door:

```
\definieersynoniemen[afkorting][afkortingen][\voluit]  
\stelsynoniemenin[letter=kap]
```

Vervolgens kan het commando `\afkorting` worden gebruikt om afkortingen te definiëren:

Verwijzen naar tekstelementen

```
\afkorting{ANWB}{Algemene Nederlandse Wielrijders Bond}  
\afkorting{VWV}{Vereniging voor Vreemdelingen Verkeer}  
\afkorting{NS}{Nederlandse Spoorwegen}
```

Als u typt:

```
De \VWV\ (\volut{\VWV}) kan u voorzien van alle toeristische  
informatie over Hasselt.
```

Dan krijgt u dit:

De vwv (Vereniging voor Vreemdelingen Verkeer) kan u voorzien van alle toeristische informatie over Hasselt.

De lijsten met synoniemen of afkortingen kunnen het best worden gedefinieerd in het instelgebied van de invoerfile. U kunt dergelijke informatie ook opslaan in een externe file (afkort.tex) die u vervolgens apart laadt:

```
\input afkort.tex (zie paragraaf 37.12)
```

Indien u een lijst met afkortingen in uw document wilt opnemen, typt u:

```
\plaatslijstmetafkortingen
```

of

```
\volledigelijstmetafkortingen
```

Een complete en gesorteerde lijst met afkortingen wordt vervolgens gegenereerd.

De weergave van lijsten kan worden beïnvloed met:

```
\stelsynoniemenin
```

In het hoofdstuk 9 vindt u een andere toepassing van synoniemen.

25 | Verwijzen naar tekstelementen

Om te verwijzen van de ene naar de andere plaats in het document kan het volgende commando worden gebruikt:

```
\in{.1.}{.2.}[ref]
```

Verwijzen naar tekstelementen

De accolades bevatten tekst, de haken bevatten een logische naam (een label). Als u in een document een volgend hoofdstuk hebt gedefinieerd:

```
\hoofdstuk[hotel]{Hotels in Hasselt}
```

dan kunt u naar dat hoofdstuk verwijzen met:

```
\in{hoofdstuk}[hotel]
```

Na verwerking is het hoofdstuknummer beschikbaar en de verwijzing zou er als volgt uit kunnen zien: *hoofdstuk 23*. Het commando `\in` wordt gebruikt voor allerlei verwijzingen naar hoofdstukken, paragrafen, figuren, tabellen, formules enz.

Een ander voorbeeld:

Er is een aantal dingen dat u in Hasselt kunt doen:

```
\startopsomming[n,opelkaar]
```

```
\som zwemmen
```

```
\som zeilen
```

```
\som[wandelen] wandelen
```

```
\som fietsen
```

```
\stopopsomming
```

Activiteiten als `\in{activiteit}[wandelen]` zijn erg vermoeiend.

Dit komt er als volgt uit te zien:

Er is een aantal dingen dat u in Hasselt kunt doen:

1. zwemmen
2. zeilen
3. wandelen
4. fietsen

Activiteiten als activiteit 3 zijn erg vermoeiend.

Het is mogelijk om naar pagina's te verwijzen met:

```
\op{.1.}{.2.}[ref]
```

Bijvoorbeeld met:

```
\op{pagina}[wandelen]
```

Dit commando wordt veel gebruikt in combinatie met:

```
\paginareferentie[ref]
```

en


```
\tekstreferentie[ref]{...}
```

Als u naar het hoofdstuk *Hotels in Hasselt* en de bijbehorende bladzijde wilt verwijzen, kunt u bijvoorbeeld de volgende tekst invoeren:

```
Kijk in \in{hoofdstuk}[hotel] op \op{pagina}[hotel] voor een  
compleet overzicht van accomodaties in  
\paginareferentie[accomodaties]Hasselt.
```

Er worden een hoofdstuknummer en een paginanummer gegenereerd bij het verwerken van de invoerfile. Op andere plaatsen in het document kan naar de locatie accomodaties worden verwezen met `\op{pagina}[accomodaties]`.

Het is ook toegestaan om een serie labels te definiëren:

```
\plaatsfiguur  
[hier]  
[fig:eengracht,fig:eenboot]  
{Een karakteristiek plaatje van Hasselt.}  
{\externfiguur[hass08g][kader=aan,breedte=5cm]}
```

Er zijn veel grachten in Hasselt (zie `\in{figuur}[fig:eengracht]`).

.
. .
.

Boten mogen gewoon in de gracht aanleggen (zie `\in{figuur}[fig:eenboot]`).

Dit wordt:



Figuur 25.1 Een karakteristiek plaatje van Hasselt.

Er zijn veel grachten in Hasselt (zie figuur 25.1). . . . Boten mogen gewoon in de gracht aanleggen (zie figuur 25.1).