PostScript en LATEX, de komplimentariteit in praktijk

Michel Goossens

CERN, CN Division
CH121 Geneva 23, Switzerland
`goossens@cernvm.cern.ch`

Abstract

In dit artikel toon ik aan hoe PostScript en LATEX een hoge graad van samenhorigheid bezitten, die het mogelijk maakt om de voordelen van beide systemen te kombineren om dokumenten elektronisch te publiceren. Allereerst vertel ik hoe, samen met de dvi-vertaler `dvips` en het stijlbestand `epsfig`, het invoegen van PostScript materiaal in een (LA)TEX bestand heel eenvoudig wordt. Samen met de stijl `rotating` kan men bijna elk gewenst globaal grafisch effekt verkrijgen zonder per-se een PostScript guru te zijn. In het tweede gedeelte van het artikel geef ik een kort overzicht van enkele op PostScript gebaseerde stijlbestanden, die bepaalde nuttige visuele effekten genereren, zoals grijze raampjes, kleurentypografie en het overdrukken van tekst. In het laatste gedeelte toon ik hoe eenvoudig het is om PostScript fonts te gebruiken in LATEX met het nieuwe fontselektie systeem (NFSS) van Frank Mittelbach.

Een woord vooaf

Dit artikel is een verkorte uit het Engels vertaalde versie van een hoofdstuk uit *het boek 'A LATEX Companion'*, dat ik samen met Frank Mittelbach (de leider van het LATEX3 project) en Alexander Samarin (ISO, Internationale Standaards Organisatie, Genève), heb geschreven, en dat in september door Addison-Wesley wordt uitgegeven. *Dit boek zal in ongeveer 400 bladzijden een overzicht geven van al wat een gevorderd LATEX gebruiker nodig heeft om haar/zijn LATEX dokument in de gewenste struktuur en vorm te gieten.* Kort samengevat ziet de inhoudstabel er als volgt uit: na een kort overzicht van wat LATEX (wel en niet) is behandelen we de strukturele basisbegrippen (hoofdstukken en paragrafen, lijsten, verbatim omgevingen, verschillende vormen van noten, meerdere kolommen, de kop- en voetteksten, drijvend materiaal). Een volledig hoofdstuk is gewijd aan tabellen en aan het nieuwe fontselektie systeem (NFSS) waarvan weldra de tweede, erg uitgebreide versie, uitkomt. Vervolgens bespreken we de mogelijkheden op het gebied van wiskundig tijpzetten met AMSLATEX en behandelen het probleem van grafisch materiaal, eerst op een drukker-onafhankelijke manier (`picture` omgeving en de `epic` en `eepic` stijlen) en dan met PostScript (zoals in dit artikel). De volgende hoofdstukken bevatten een gedetailleerde bespreking van hoe men een zakenregister (index) genereert met *MakeIndex* en hoe men literatuurlijsten en bibliografische gegevensbanken creëert en beheert met BibTEX. Het laatste hoofstuk vertelt hoe men zijn LATEX stijlen kan dokumenteren en verspreiden met de stijl `doc` en docstrip. De eerste appendix beschrijft waar men terecht kan op het internet of via elektronische post om LATEX stijlbestanden en informatie te verkrijgen, terwijl de tweede appendix een gedetailleerd overzicht geeft van LATEX's gevorderde begrippen (het definiëren van opdrachten en omgevingen, het gebruik van tellers en lengtes, de konstruktie en het nut van raampjes en tenslotte hoe men met LATEX arithmetiek bedrijft).

De helft van de opbrengst van dit boek gaat direkt naar de financiering van het LATEX3 projekt, zodat U, door het kopen van dit boek, zich niet alleen een handig referentiewerk aanschaft, maar ook ervoor zorgt dat LATEX in de toekomst nóg beter wordt.

1 Het samenbrengen van tekst en PostScript grafisch materiaal

In dit artikel beschrijf ik enkele LATEX stijlen die, samen met Tom Rokicki's `dvips` [1] (en waarschijnlijk ook andere dvi-vertaalprogramma's) het mogelijk maken PostScript's uitgebreid opdrachtenpalet [2, 3, 4, 5, 6, 7] te gebruiken om bepaalde grafische effekten te verkrijgen. Het enige waar U voor moet opletten indien U gebruik wilt maken van deze functies, is dat hun effekt slechts zichtbaar wordt indien U ook beschikt over een PostScript drukker of visualizator, een programma dat PostScript kan omzetten in een beeld op het scherm van een werkstation of een persoonlijke computer (PC of Mac). Een voorbeeld van zo'n programma in het publieke domein is ghostview (dat deel uitmaakt van het GNU software projekt).

Niettegenstaande het feit dat het mogelijk is om via de TEX opdracht `\special` willekeurige teksten door te spelen aan de dvi-vertaalprogramma's (in ons geval elementaire PostScript opdrachten), is het in het algemeen aan te raden meer generische konstrukties te gebruiken. De eerste stijl, die ik bespreek, heet `epsfig` en maakt het U eenvoudig om een PostScript beeld in te voeren in een LaTeX bestand. Zoals de naam het aanduidt gaat het hier over een 'Encapsulated PostScript' (EPS) [8] bestand d.w.z. een bestand dat, o.a. informatie bevat over zijn dimensies, meer in het bijzonder een regel bevat met de struktuur:

```
%%BoundingBox: llx lly urx ury
```

met `llx` en `lly` de x- en y-dimensies (in PostScript punten, waar 1 PostScript punt $= 0,35278$mm) van de linker beneden hoek en `urx` en `ury` de x- en y-dimensies van de rechter bovenhoek van een (virtuele) rechthoek, die het gehele beeld omsluit. Aan de hand van deze dimensies kan de opdracht `epsfig` de plaats berekenen, die opengelaten moet worden om het beeld te kunnen afdrukken. Kleine beeldjes, zoals '  ' en '⊗' kunnen om het even overal in het document worden gebruikt.
De volledige syntax van de opdracht is:

```
\epsfig{file=fn,%
        height=ht,width=wd,%
        clip=,rotate=angle,silent,%
        bbllx=blx,bblly=bly,%
        bburx=brx,bbury=bry}
```

`file` De naam van het Encapsulated PostScript bestand (je kan ook `figure=` gebruiken).

`height` De gewenste hoogte van het beeld op het gedrukt blad (in één van de TeX eenheden). Indien deze parameter en de volgende (`width`) niet gespecificeerd zijn dan zal het beeld zijn 'natuurlijke' grootte hebben, d.w.z. de dimensies gespecificeerd op de `BoundingBox` lijn in het Encapsulated PostScript bestand zelf. Waneer men de breedte (`width=`) specificeert en geen hoogte (`height=`), dan wordt deze laatste automatisch aangepast zodanig dat de verhouding tussen beide dimensies onveranderd blijft.

`width` De gewenste breedte van het beeld op het gedrukt blad (in één van de TeX eenheden). Indien deze parameter en de vorige (`height`) niet gespecificeerd zijn dan zal het beeld zijn 'natuurlijke' grootte hebben, d.w.z. de dimensies gespecificeerd op de `BoundingBox` lijn in het Encapsulated PostScript bestand zelf. Waneer men de hoogte (`height=`) specificeert en geen breedte (`width=`), dan wordt deze laatste automatisch aangepast zodanig dat de verhouding tussen beide dimensies onveranderd blijft.

`clip` Wanneer deze optie wordt gekozen, dan worden de delen van het beeld die buiten de 'BoundingBox' vallen, weggeknipt. Opgelet dat `"clip="` geen waarde neemt, maar dat het '='-teken aanwezig moet zijn.

`rotate` De draaihoek (in graden, de positieve richting gaat tegen de richting van de wijzers van een uurwerk in) voor het gehele beeld.

`silent` Met deze optie werkt `\epsfig` 'stilletjes', d.w.z. zonder informatie op het beeldscherm (of in de log) te schrijven.

`bbllx` De **x**-coördinaat van de linker benedenhoek van de omschreven rechthoek (BoundingBox).

`bblly` De **y**-coördinaat van de linker benedenhoek van de omschreven rechthoek (BoundingBox).

`bburx` De **x**-coördinaat van de rechter bovenhoek van de omschreven rechthoek (BoundingBox).

`bbury` De **y**-coördinaat van de rechter bovenhoek van de omschreven rechthoek (BoundingBox).

Wanneer de BoundingBox parameters **vóór** de naam van de figuur staan (`figure=` or `file=`), dan negeert `\epsfig` de parameters in het Encapsulated PostScript bestand (`\epsfig` zal dan zelfs het bewuste bestand niet trachten te lezen), en gebruikt de dimensies gespecifieerd in de opdracht zelf. Dat kan interessant zijn om een bepaald gedeelte van het beeld weg te knippen (clipping), of om PostScript bestanden te gebruiken die de BoundingBox informatie niet bevatten of waar deze laatste verkeerd is. U moet wel oppassen dat u geen spaties laat in de argumenten van `\epsfig`, want dan werkt die opdracht niet; als U verplicht bent een `\epsfig` opdracht over verschillende regels uit te schrijven, dan moet U elke lijn afsluiten met een %-teken (zoals ik ook deed in de beschrijving van de opdracht hierboven).

1.1 Simpele figuren

De eenvoudigste situatie is wanneer U tevreden bent met de 'natuurlijke' dimensies van het beeld (zoals gespecificeerd in het bestand zelf op de `Bounding-Box` lijn). Dan heeft U geen schaalfaktor nodig en `\epsfig` berekent de plaats die het plaatje inneemt eenvoudig uitgaande van deze dimensies. Een eenvoudige manier om de natuurlijke dimensies van een beeld te kennen is het te drukken op een PostScript printer. De linker benedenhoek van het beeld wordt gepositioneerd daar waar de `\epsfig` opdracht wordt gegeven. Indien de opgegeven breedte (`width`) of hoogte (`height`) niet overeenstemmen met de respektieve natuurlijke dimensies, dan word het beeld met de nodige schaalfaktor vergroot of verkleind.

Figuur 1 op pagina 104 is een prent waarvoor we een hoogte van 5cm wensen. De `\epsfig` opdracht werd in een `\mbox` gestopt om de figuur te centreren met behulp van een `center` omgeving.

1.2 Proeffiguren

Sommige PostScript figuren hebben een lange tijd nodig om gedrukt te worden. Voor zulke figuren is er een 'proef' (draft) optie, die niet het PostScript beeld zelf afdrukt, maar ze vervangt door een rechthoek met dezelfde dimensies als het uiteindelijke beeld in de tekst zelf, tezamen met de naam van de figuur (zie figuur 2 op pagina 104). De opdracht `\psdraft` schakelt om naar proefmode, zodat alle volgende `epsfig` opdrachten 'draft' rechthoekjes, in plaats van de oorspronkelijke figuren, zullen drukken. De opdracht `\psfull`, keert terug naar de normale mode. Een ander voordeel van de draft-mode is dat er geen TeX `\special` opdrachten in het dvi-bestand worden geschreven, zodat het test dokument bekeken kan worden met een willekeurig visualizatie-programma, dat niet noodzakelijk PostScript behoeft te verstaan.

1.3 Schikking van beeldmateriaal

We kunnen nu de mogelijkheden van de `\epsfig` opdracht en van LaTeX kombineren om ons beeldmateriaal optimaal op het blad te schikken. De figuren 3 tot 5 op pagina 119 tonen schikkingen, verkregen door gebruik te maken van `minipage` omgevingen.

2 Het roteren van tekst- en beeldmateriaal

Het komt vaak voor dat we een figuur of tabel willen draaien, enerzijds om de presentatie van de informatie te verbeteren (bijvoorbeeld een tabel wijder dan de kolombreedte), anderzijds om bepaalde visuele effekten te bereiken. Een handige manier om deze draaifunktionaliteit te verkrijgen is met de stijl `rotating`, ontwikkeld door Sebastian Rahtz en Leonor Barroca [9], die enkele nieuwe LaTeX omgevingen definiëert die ik hierna één voor één onder de loep wil nemen.

2.1 Het roteren van tekst

De `rotate` omgeving draait het omsloten materiaal over het opgegeven aantal graden (positief is met de wijzers mee, zoals in PostScript). Deze omgeving neemt echter aan dat het materiaal geen plaats inneemt, zodat U deze omgeving kunt gebruiken om speciale effekten te verkrijgen. Plaats het TeX kommentaar teken % aan het einde van de opdracht `\begin{rotate}{52}%` — U moet de opdracht altijd volgen met tekst of met %, om te voorkomen that ongewenste spaties in de tekst opduiken.

```
Begin van de zin \begin{rotate}{-52}%
   NTG
\end{rotate} Einde van de zin
```

Begin van de zin NTG Einde van de zin

Indien U wenst dat LaTeX de nodige ruimte openlaat om het materiaal te plaatsen, dan gebruikt U de omgeving `turn`:

```
Begin van de zin \begin{turn}{52}%
   NTG
\end{turn} Einde van de zin
```

Begin van de zin NTG Einde van de zin

De omgeving '`sideways`' is een speciaal geval, van de `turn` omgeving, met een draaihoek van -90:

```
Begin van de zin \begin{sideways}%
   NTG
\end{sideways} Einde van de zin
```

Begin van de zin NTG Einde van de zin

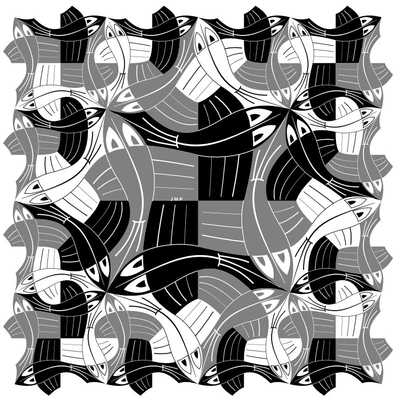
Wanneer men met hele paragrafen tekst werkt, dan vindt men dat TeX ramen ('boxes') in het algemeen zowel een hoogte als een diepte hebben. In dit geval vinden de rotaties plaats rond het uiterste linker punt waar het raam de basislijn raakt. Dit kan soms tot eigenaardige en onverwachte resultaten leiden, zoals figuur 6 op pagina 120 toont. Voor elk beeldje tonen de lijntjes links en rechts de positie van de basislijn. U kunt de positionering van de ramen min of meer kontroleren door het specificeren van de (optionele) positie parameter die beschikbaar zijn bij bepaalde LaTeX opdrachten.

```
\newcommand{\B}{A b C d E f G h I j K
                l M n O p Q}
Begin    \begin{turn}{45}
           \parbox[t]{16mm}{\B}
         \end{turn}
Midden   \begin{turn}{45}
           \parbox[b]{16mm}{\B}
         \end{turn}
Einde
```

Begin A b C d E f G h I j K l M n O p Q Midden A b C d E f G h I j K l M n O p Q Einde

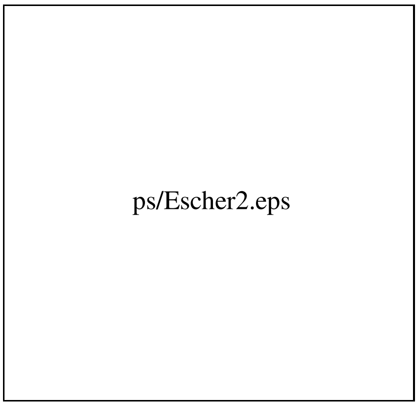
2.2 Het roteren van tabellen

Materiaal in tabellen (LaTeX's `tabular` en `array` omgevingen) kunnen op een analoge manier gedraaid worden. De voorbeelden in figuur 7 op pagina 120 tonen hoe 'onzichtbare regels' gebruikt worden om de hoogte of de breedte van de kolommen te kontroleren.



Figuur 1: *Een eenvoudig gecentreerde figuur*

```
\begin{center}
 \mbox{\epsfig{file=ps/Escher2.eps,height=50mm}}
\end{center}
\caption[]{Een eenvoudig gecentreerde figuur}
\label{fig:simple}
```



Figuur 2: *Een figuur in test (draft) mode*

```
% ga naar test (draft) mode
 \psdraft

 \begin{center}
  \mbox{\epsfig{file=ps/Escher2.eps,height=50mm}}
 \end{center}
 \caption[]{Een figuur in test (draft) mode}
 \label{fig:draft}
% ga terug naar de normale mode
 \psfull
```

Rotaties kunnen gekombineerd worden, als volgt:

```
\begin{sideways}
\begin{tabular}{l@{\qquad}r}
\em Zin \rule{0pt}{3cm}
            & \begin{rotate}{-90}%
Frequentie\end{rotate}  \\[1mm]
 \hline
 Goeie dag & 33\\
 Tot ziens & 34\\
 \hline
\end{tabular}
\end{sideways}
```

Zin	Frequentie
Goeie dag	33
Tot ziens	34

Een ingewikkelder (en wellicht meer voorkomend) voorbeeld is tabel 1 op pagina 105, die een `sideways` omgeving in een `sidewaystable` omgeving toont. Deze laatste omgeving draait niet enkel de tabel zelf, maar ook zijn onderschrift. De `sidewaystable` omgeving — en zijn analoog voor figuren, de `sidewaysfigure` omgeving — nemen echter een volledig pagina in, en men moet dus wel een beetje uitkijken om zeker te zijn dat men de pagina wel degelijk op een aanvaardbare manier kan vullen.

	gebruikersprijzen		**produktieprijzen**		**werkloosheid %**	
	3 maanden	**1 jaar**	**3 maanden**	**1 jaar**	**laatste**	**1 jaar**
Australië	+0,4	+0,8	+4,6	+2,1	11,4	10,5
België	+1.9	+2,2	−3,0	−0,7	8,4	7,7
Canada	+1,0	+1,7	+3,3	+2,9	11,8	10,3
Duitsland	+3,2	+3,7	−0,8	+0,5	7,1	6,3
Frankrijk	+1,4	+2,1	−1,9	−0,9	10,4	9,9
Groot Brittanië	+1,9	+3,0	+1,4	+3,3	10,3	8,9
Italië	+4,2	+4,8	−2,1	+1,0	10,6	10,3
Japan	+0,4	+1,1	−2,6	−1,3	2,2	2,0
Nederland	+5,1	+2,9	−2,1	+0,1	4,5	4,4
Spanje	+6,6	+5,1	+0,2	+1,2	15,3	15,4
Verenigde Staten	+3,4	+3,0	+1,4	+1,2	7,2	6,9
Zweden	+5,7	+1,3	−3,7	−1,3	5,4	3,2
Zwitserland	+2,4	+3,3	−0,6	−0,1	3,9	1,7

```
renewcommand{\arraystretch}{1.2}
\begin{sideways}
\begin{tabular}{@{}>{\bf}l@{\qquad}%
  >{$}c<{$}@{\quad}>{$}r<{$}@{\qquad}%
  >{$}c<{$}@{\quad}>{$}r<{$}@{\qquad}%
  >{$}r<{$}@{\quad}>{$}r<{$}}
 & \multicolumn{2}{c}{\textbf{gebruikerspr"yzen}}
 & \multicolumn{2}{c}{\textbf{produktiepr"yzen}}
 & \multicolumn{2}{c}{\textbf{werkloosheid \%}}\\
 & \multicolumn{1}{c}{\textbf{3 maanden}}
 & \multicolumn{1}{r}{\textbf{1 jaar}}
 & \multicolumn{1}{c}{\textbf{3 maanden}}
 & \multicolumn{1}{r}{\textbf{1 jaar}}
 & \multicolumn{1}{c}{\textbf{laatste}}
 & \multicolumn{1}{r}{\textbf{1 jaar}}    \\\hline
Australi"e &+0,4&+0,8&+4,6&+2,1&11,4&10,5 \\\hline
Belgi"e    &+1.9&+2,2&-3,0&-0,7& 8,4& 7,7 \\\hline
Canada     &+1,0&+1,7&+3,3&+2,9&11,8&10,3 \\\hline
Duitsland  &+3,2&+3,7&-0,8&+0,5& 7,1& 6,3 \\\hline
Frankr"yk  &+1,4&+2,1&-1,9&-0,9&10,4& 9,9 \\\hline
Groot Brittani"e
           &+1,9&+3,0&+1,4&+3,3&10,3& 8,9 \\\hline
Itali"e    &+4,2&+4,8&-2,1&+1,0&10,6&10,3 \\\hline
Japan      &+0,4&+1,1&-2,6&-1,3& 2,2& 2,0 \\\hline
Nederland  &+5,1&+2,9&-2,1&+0,1& 4,5& 4,4 \\\hline
Spanje     &+6,6&+5,1&+0,2&+1,2&15,3&15,4 \\\hline
Verenigde Staten
           &+3,4&+3,0&+1,4&+1,2& 7,2& 6,9 \\\hline
Zweden     &+5,7&+1,3&-3,7&-1,3& 5,4& 3,2 \\\hline
Zwitserland&+2,4&+3,3&-0,6&-0,1& 3,9& 1,7 \\\hline
\end{tabular}
\end{sideways}
```

Tabel 1: *Hoe men een tabel kan draaien*

2.3 Het roteren van figuren

Met de `rotating` stijl kunt U niet enkel tekst maar ook PostScript beeldmateriaal draaien. Een voorbeeld ziet U in figuur 8 op pagina 121, waar een Encapsulated PostScript beeld in het dokument wordt ingevoerd met de `\epsfig` opdracht, en daarna wordt gedraaid naar keuze. Het is interessant om voor elk geval naar de positie van de basislijn te kijken.

3 Het kleuren van ramen

Om de nadruk te leggen op een bepaald gedeelte van de tekst is het dikwijls nuttig het bedoelde materiaal te omlijnen of het met een grijze achtergrond te kleuren.

De stijl `psboxit` (geschreven door J. Maillot) bouwt een PostScript raam boven op een TEX raam, gebruik makend van de parameters berekend voor deze laatste. Om de mogelijkheden van deze stijl te benutten moeten de PostScript opdrachten geïnitialiseerd worden, door de opdracht `\PScommands` te geven aan het begin van het dokument.

De basisopdracht is `\psboxit`, gedefiniëerd as volgt:

`\psboxit`{*PS opdrachten*}{*TEX materiaal*}

Deze opdracht onderwerpt het materiaal gespecificeerd in het tweede argument '*TEX materiaal*' aan het effekt van de PostScript opdrachten in het eerste argument '*PS opdrachten*'. Het stijlbestand definiëert verscheidene PostScript proceduren, zoals `cartouche` en `rect-cartouche`. Deze laten toe zekere visuele effekten te verkrijgen, zoals getoond in het voorbeeld hieronder:

```
--\psboxit{5 cartouche}{NTG}--
--\psboxit{rectcartouche}{\spbox{DANTE}}-
-
--\psboxit{box .7 setgray fill}%
          {\spbox{TUG}}--
```

–NTG– – DANTE – – TUG –

De bijkomende opdracht `\spbox` werkt zoals `\fbox`, d.w.z. men verkrijgt een raam rond het ingeschreven materiaal met een afstand `\fboxsep` toegevoegd in elke richting. Het enige verschil is dat `\spbox` geen lijnen afbeeldt rond het raam. Het belangrijkste doel van `\fboxsep` is raam te bouwen met identieke afmetingen als `\fbox`, maar met een gekleurde achtergrond. In het algemeen kunt U natuurlijk zelf opdrachten definiëren, zoals:

```
\newcommand{\grijsraam}[1]%
{\psboxit{box .7 setgray fill}{\fbox{#1}}}
```

Indien U grotere ramen wilt kleuren, dan gebruikt U het best de `boxitpara` omgeving:

```
\begin{boxitpara}{box 0.7 setgray fill}
Amsterdam is een grote stad.
Antwerpen heeft een mooie kathedraal.
Groningen ligt in het noorden.
Leuven ligt in het zuiden.
\end{boxitpara}
```

Amsterdam is een grote stad. Antwerpen heeft een mooie kathedraal. Groningen ligt in het noorden. Leuven ligt in het zuiden.

3.1 Werken met kleuren

Echte kleuren, in tegenstelling met verschillende grijsniveau's, zijn natuurlijk enkel te verkrijgen wanneer men over een kleurendukker beschikt. `dvips` maakt het mogelijk om PostScript's kleurenopdrachten te gebruiken, en stelt hiervoor de stijl `dpscolor` en `dpsblack` ter beschikking. Deze laatste stijl neutraliseert alle PostScript kleurenopdrachten van `dpscolor`, zodat men een bestand dat deze opdrachten bevat, zonder problemen kan afdrukken op een zwartwit printer. `dpscolor` biedt een pallet van 68 kleuren (de komplete lijst vindt U in het stijlbestand zelf).

De kleur van de achtergrond voor de lopende en volgende bladzijden wordt bepaald door de opdracht `\background`, met als parameter de naam van de kleur. Indien men een groene achtergrond wenst dan zegt men:

```
\background{Green}
```

Wanneer men dan terug wilt naar de gewone (default) witte achtergrond, dan geeft men:

```
\background{White}
```

`dpscolor` heeft twee soorten kleuropdrachten; de eerste heeft één argument, om korte teksten te kleuren, bijvoorbeeld:

```
\Blue{De volgende tekst wordt in 't
      blauw gezet.}
```

De volgende tekst wordt in 't blauw gezet.

terwijl de tweede de kleur voor de tekst herdefiniëert, en deze definitie onveranderd laat tot aan de volgende kleuropdracht.

```
\textGreen De volgende tekst is in
  't groen. En dat bl"yft ie tot we
  de kleur herdefini"eren.
\textBlack De tekst wordt terug zwart
```

De volgende tekst is in 't groen. En dat blijft ie tot we de kleur herdefiniëren. De tekst wordt terug zwart

Bijkomende kleuren kunnen gedefiniëerd worden in funktie van het CMYK (Cyan, Magenta, Yellow, blacK) kleurenmodel, waarin men een kleur specifiëert aan de hand van de intensiteit (een getal tussen 0.0 en 1.0) van elk van haar vier kleurenkomponenten.

```
\Color{.2 .3 .4 .5}{De kleur van deze
       tekst werd gespecifi"eerd  als
       een CMYK kwadruplet}
```

De kleur van deze tekst werd gespecifiëerd als een CMYK kwadruplet