

T_EX zonder omhaal voor Atari ST en andere PC's (VI)

— aan het werk met METAFONT —

Rens Swart

De Blickestraat 199, 3572 WC Utrecht

Abstract

METAFONT is een programma waarmee het mogelijk is om via wiskundig beschreven curves en oppervlakken lettertypes of andere symbolen te ontwerpen. Deze kunnen gebruikt worden in het zetprogramma T_EX.

Dit artikel gaat niet in op het werkelijke ontwerp-proces en zelfs nauwelijks op de faciliteiten die METAFONT hiertoe biedt, maar op de vraag hoe METAFONT functioneert in de Atari-omgeving, gebaseerd op het Public Domain pakket van Christoph Strunk.¹ Kortom: hoe kun je aan het werk met METAFONT?

Deze cursus is een herdruk van een serie artikelen uit het blad ST², uitgegeven door Stichting ST³. *Daar zijn de in de tekst genoemde schijffjes te verkrijgen.*

1 Inleiding

METAFONT en T_EX zijn nauw met elkaar verwant. Beide zijn ontworpen door een hoogleraar Wiskunde en Informatica van de universiteit van Stanford in de Verenigde Staten van Amerika, Donald Knuth. Deze ergerde zich zo aan de typografische kwaliteit van zijn boeken dat hij eind zeventiger jaren besloot programmatuur te ontwikkelen die het wetenschappers mogelijk moest maken zelf hun artikelen en boeken te zetten. Daarbij wilde hij de typografische kennis en het grafische inzicht van ambachtelijke zettters in het pakket inbouwen.

Er ontstonden twee pakketten: T_EX om teksten te zetten en METAFONT om hiervoor de lettertekens te ontwerpen. T_EX is een geavanceerd programma vol typografische kennis, dat bij de opmaak van een regel niet alleen naar die regel kijkt maar ook naar de consequenties voor de gehele bladzijde, hetgeen de kans op typografisch niet geheel bevredigende uitvoer verkleint. Het programma werkt daarom ook niet interactief: een tekst moet worden voorzien van opdracht-codes en vervolgens door een soort vertaler gehaald worden. Met name het automatisch zo fraai mogelijk zetten van formules is de sterke kant van T_EX. Het gebruik van het pakket is in de Wiskunde en Natuurwetenschappen wereldwijd zeer verbreid.

Knuth heeft voor het gebruik in T_EX met zijn programma METAFONT een familie lettertypes gemaakt die hij Computer Modern noemt. De tienpunts romein bijvoorbeeld heet daarom cmr10, de 8 punts bold extended (breed vet) heet cmbx8. Het gebruik van METAFONT om lettertypes te maken heeft lang niet zo om zich heen gegrepen als het gebruik van T_EX, zodat cmr algemeen gebruikt wordt.

Werken met METAFONT is geheel anders dan werken met T_EX. Is voor het eenvoudigste gebruik van T_EX nauwelijks kennis nodig en kunnen met enkele opdrachten al fraai gezette teksten gemaakt worden, METAFONT moet men eigenlijk zien als een programmeertaal, waarmee alleen goede resultaten bereikt kunnen worden als men hem na een gedegen studie enigszins in de vingers heeft zitten. Het METAFONTboek van Donald Knuth (uiteraard opgemaakt met T_EX) is onmisbaar.⁴ En dan nog komen daarin alleen de faciliteiten van METAFONT aan bod, niet het ontwerpen zelf. Het ontwerpen van lettertypes is een vak apart en vergt veel tijd en goede smaak.

2 Wat doet Metafont?

Om het procedé van het ontwerpen en beoordelen van een lettertype in METAFONT te begrijpen is het goed eerst te schetsen wat er gebeurt als een kant-en-klaar ontwerp door METAFONT gehaald wordt. Een METAFONT-bestand bevat voor ieder teken in het lettertype een soort programma dat de vorm van het teken, de te gebruiken denkbeeldige pennen en de weg die met elke pen moet worden afgelegd beschrijft. METAFONT kijkt welke pixels hierdoor uiteindelijk zwart worden en schrijft dit pixelbeeld weg.

METAFONT rekent zelf in pixels. Omdat het onprettig zou zijn als een letter op een Deskjet-afdrukker met 300 pixels per inch driemaal zo klein zou zijn als op het scherm met 96 pixels per inch, kan METAFONT worden megedeeld hoeveel millimeter een letter moet zijn. METAFONT rekent dan zelf uit hoeveel pixels dat is, afhankelijk van het uitvoerapparaat. Daartoe zijn in de *plain base* (de standaard

¹ Verkrijgbaar in het public domain bij de Stichting ST, schijven B95 t/m 105.

² Uitgave ST51.

³ Stichting ST, Postbus 11129, 2301 EC Leiden, tel. 071-130045. ST is een onafhankelijk tijdschrift van en voor gebruikers van Atari ST computers.

⁴ The METAFONTbook, Donald E. Knuth, Addison-Wesley, ISBN 0-201-13444-6, prijs omstreeks f 65,-.

macroverzameling van METAFONT) een aantal modi gedefinieerd waarmee onder andere het aantal pixels per inch kan worden gegeven. Zo is er een *mode=atari* (voor het scherm) en een *mode=deskjet*, die we in het vervolg zullen gebruiken als we het hebben over het maken van lettertypes voor de afdrucker. Later gaan we verder op het begrip modus in.

Van het *.mf*-bestand met het lettertype maakt METAFONT een *generic font*-bestand *.gf*. Dit kan worden gecomprimeerd met het programma *gftopk*, dat een *.pk*-bestand (*packed*) oplevert. Hierin wordt elk teken exact in pixels beschreven. Tegelijk levert METAFONT een *.tfm*-bestand (*T_EX font metric data*), waarin de diverse afmetingen van elk teken worden beschreven. Wordt nu met T_EX een tekst gezet uit dit vers gegenereerde lettertype, dan gebruikt T_EX het *.tfm*-bestand om te kijken waar de tekens precies moeten komen. T_EX hoeft tenslotte niet te weten waar de pixels komen, alleen hoe de letters tegen elkaar gezet moeten worden en dat is alleen afhankelijk van hun afmetingen. T_EX maakt een *.dvi*-bestand (device independent, apparaat-onafhankelijk) waarin dat staat. Ten slotte is er nog een programma nodig dat aan de hand van het *.dvi*-bestand de pixels werkelijk op de plek van elk teken zet. Dat programma gebruikt het *.pk*-bestand, dat elk teken in het juiste aantal pixels voor het betreffende uitvoerapparaat beschrijft. Omdat dit programma het aldus in het geheugen opgebouwde veld pixels samen met de juiste grafische printercommando's naar de printer (of het scherm) moet sturen, is er voor elk uitvoerapparaat een apart programma, bijvoorbeeld *dvi_dj* voor de Deskjet inktstraaldrukker of *dvi_view* voor het scherm.

Voor elke vergroting van een lettertype — heel eenvoudig aan METAFONT mee te geven — is de beschrijving in pixels natuurlijk anders. Daarom is er voor elke vergroting een apart *.pk*-bestand. In de Atari-omgeving hebben ze exact dezelfde naam. Daarom is ervoor gekozen om ze in verschillende folders te plaatsen om ze te kunnen onderscheiden. In de foldernaam is de vergroting opgenomen.⁵ Dit gedoe met folders neemt METAFONT niet zelf voor zijn rekening: dat doet de *shell* *c_mf.prg*, die de onvolprezen Christoph Strunk er voor de Atari omheen gemaakt heeft. Voor T_EX maakt een vergroting niet veel uit, omdat het niets met pixels te maken heeft: het berekent gewoon de vergrote afmetingen van elke letter uit het *.tfm*-bestand.

3 Basisgebruik Metafont

Doorgaans zal men METAFONT alleen gebruiken om met behulp van bestaande lettertype-ontwerpen lettertypes aan te maken op de gewenste vergrotingen en voor de gewenste uitvoerapparatuur. Laten we even kijken hoe dat gaat. In het vervolg ga ik ervan uit dat u het complete public domain T_EX-pakket van de Stichting ST hebt en dat u dat

geïnstalleerd hebt op harde schijf. Voor METAFONT moeten schijven B102 en B103 geheel zijn geïnstalleerd. Lees bestand *info* op die schijven en kijk eventueel naar de bestanden in *\tex\tex_doc*, met name naar *st_info.txt*.

De METAFONT-programmatuur bestaat uit diverse grote programma's, die zelfstandig te gebruiken zijn. Eenvoudig is dat echter niet. De bewerker van T_EX en METAFONT voor de Atari, Christoph Strunk, heeft een grafische schil om deze programmatuur heen geschreven: *c_mf.prg*. Vanuit dit programma laat bijna alles zich eenvoudig met de muis sturen. Na automatische installatie staan de meeste parameters al goed. Bekijk bijvoorbeeld menukeuze *Environment* onder *Parameter* en lees eventueel genoemde *st_info.txt*. De shell moet elk van de programma's kunnen vinden: daartoe dient menu *Finden*. Onder *Arbeiten* wordt het proces werkelijk in gang gezet. Daar staat ook een *externes Programm*: handig is bijvoorbeeld een editor, die onder *Finden* wel eerst moet worden aangemeld. Het uitvoerapparaat waarvoor moet worden gegenereerd laat zich eenvoudig onder *Device* aanklikken. De vergroting laat zich eveneens makkelijk kiezen.

Nu worden de gewenste lettertypes aangeklikt, de gewenste vergroting ingesteld en het apparaat waarvoor de lettertypes moeten worden aangemaakt. De schil kan zelf nagaan of lettertypes al aanwezig zijn en zal dat melden of ze direct aanmaken. Welke van alle getoonde lettertypes ontbreken op de ingestelde vergroting kan worden getoond door *c_mf* ze te laten selecteren: kies *selektieren-fehlende* onder *Auswahl*. Veel praktischer is zelf de gewenste lettertypes aan te klikken en vervolgens *Auswahl: Testen, welche fehlen* onder *Arbeiten* te kiezen, of *Fehlende erzeugen*, waarna de ontbrekende terstond worden gemaakt.

De schil kan ook gebruikmaken van bestanden met lettertype-lijsten. Een aantal staat in de folder *\metafont\fts*. Lijsten hebben, behalve dat men zelf een voorkeurselectie kan maken, als voordeel dat verschillende vergrotingen kunnen worden aangegeven: bij aanklikken (*Auswahl*) kan dat niet. De schil controleert na selectie van *Liste: Fehlende erzeugen* onder *Arbeiten* zelf welke lettertypes nog niet aanwezig zijn. Zelfs de voor de vergrotingen benodigde folders worden automatisch gemaakt.

Het is ook mogelijk vanuit T_EX de nog ontbrekende lettertypes, nodig voor laten we zeggen *brief.tex*, automatisch te laten aanmaken. Nadat T_EX op *brief.tex* is losgelaten kiezen we in de T_EX-schil *ctex.prg* onder *Arbeiten* voor *Fonts erzeugen*. Uit *brief.dvi* wordt dan *brief.fts* gemaakt, waarna de METAFONT-schil gestart wordt. Nu kan (na kiezen van het juiste uitvoerapparaat) *Liste: Fehlende erzeugen* onder *Arbeiten* worden gekozen.

⁵In de Atari-omgeving bevat de folder *\tex\font_s* folders voor verschillende vergrotingen van de beeldscherm-lettertypes. De niet-vergrote lettertypes staan in folder *font0096*: 96 is het aantal pixels per inch van het scherm. Lettertypes voor een vergroting met magstep 2 (1,2 maal 1,2) staan in *font0138*: 138 is 1,44 maal 96. De bestanden zelf heten allemaal bijvoorbeeld *cmr10.pk*. Knuth gebruikt oorspronkelijk in plaats hiervan *cmr10.96gf* voor het niet vergrote lettertype en *cmr10.138gf* voor het 1,44 maal vergrote lettertype, zodat geen afzonderlijke folders nodig zijn.

Bij de standaardinstellingen van `c_mf` komen de `tfm`-bestanden terecht in `meta-font\outputs`, terwijl `ctex` ze zoekt in `\tex\tfm`. Copieer ze dus daarheen, als ze er nog niet stonden.

Eigen instellingen van de schil kunnen worden bewaard: door *Speichern* onder *Parameter* te kiezen wordt bestand `c_mf.inf` geschreven. Bij starten van `c_mf.prg` wordt dit automatisch geladen. Het is niet mogelijk meerdere sets instellingen te bewaren, tenzij door tevoren en nadien het bestandje te hernoemen.

4 Bestanden voor Metafont

De schil `c_mf` zit zo slim in elkaar dat alle benodigde bestanden worden gevonden. Wat gebeurt er bijvoorbeeld als we het standaard Computer Modern Roman lettertype op 12 punts willen hebben, maar met factor 1,2 vergroot? Na *alle deselektieren* onder *Auswahl* klikken we op `cmr12`. We kiezen *HP Deskjet* onder *Device*: daarvoor moet het lettertype worden gemaakt. Klik op *Magnification*, kies vergroting 1200. Kijk eventueel of het lettertype er al is: kies *Auswahl: Testen, welche fehlen* onder *Arbeiten*. Er volgt een keurige melding. Kies *Auswahl: Fehlende erzeugen*.

Wat METAFONT doet komt behalve op het scherm ook in een log-bestand. In *Environment* onder *Parameter* staan de *environment-variabelen* die door de schil `c_mf` worden gebruikt om de plaats van de invoerbestanden te bepalen en waarheen de uitvoer moet worden geschreven. `c_mf` bepaalt via *environment-variabele* `MFOUTPUTS` niet alleen waarheen METAFONT de `.gf`- en `.tfm`-bestanden moet schrijven, ook komt daar het logbestand te staan.

Ergens in dat bestand staat: `input cmr12`.

`c_mf` vindt dit bestandje met behulp van een *environment-variabele*, namelijk

```
MFINPUTS = .\atari;.*
```

waarbij, zoals we lezen, de punt de hoofdfolder is, meestal `metafont\inputs*.mf`. Dit is in te stellen met *Haupt-Ordner* onder *Finden*. `c_mf` is in staat tot drie niveaus diep onder deze hoofdfolder te zoeken naar invoerbestanden. In paragraaf 6 van bestand `st_info.txt` staat dat eerst in `.\atari` wordt gezocht, daarna pas in andere folders onder `metafont\inputs`. Dit heeft Strunk zo geprogrammeerd, omdat gewijzigde bestanden dan in `.\atari` kunnen worden gezet, terwijl de originelen gewoon kunnen blijven staan. `cmr12.mf` wordt uiteindelijk gevonden in `metafont\inputs\cm\param`. Het bestand bevat alleen een flink aantal parameters: het werkelijk genereren gebeurt elders. Door de `generate roman` aan het einde wordt bestand `cm\drive\roman.mf` gelezen. Dit op zijn beurt roept onder andere `cm\drive\roman1.mf` op, waarin de kleine letters staan (`l = lowercase`), maar ook `punct.mf` die onder andere punt, komma, uitroepetekens en haakjes bevat.

5 De plain base

METAFONT is eigenlijk een verzameling primitieve functies. Slechts door het laden van een verzameling macro's is er fatsoenlijk mee te werken. De standaard macroverzameling heet *plain base* en wordt door METAFONT direct geladen. Deze verzameling staat in `\metafont\bases\plain.bse`.

Normaal is dit bestand al aanwezig, maar het is ook aan te maken. Het programma `inimf.ttp` compileert als het ware het METAFONT-bestand `plain.mf` naar `plain.bse`. Zoals in `metafont\inputs\base\plain.ini` te zien is, wordt behalve `plain.mf` ook het bestand `\metafont\inputs\atari\atari.mf` in de *plain base* voor METAFONT ingebakken. In `atari.mf` staan de definities van de modi voor de diverse uitvoerapparaten. Wordt iets in dit bestand gewijzigd, dan heeft dat voor METAFONT dus alleen effect als het wordt verwerkt in een nieuwe *plain base*, dus na opnieuw draaien van `inimf.ttp`. Dit kan in `c_mf` met *Base file erzeugen* onder *Arbeiten*.

Ontwerper Knuth heeft voor zijn Computer Modern lettertypes een speciale macroverzameling gemaakt die resulteert in een speciale base file: `cm.bse`. Als de eerste twee letters van de naam van een lettertype `cm` zijn, neemt `c_mf` automatisch deze base. Dit is vastgelegd bij *BASE files* onder *Parameter*.

Wat `c_mf` doet is direct na de invoerprompt `**` van METAFONT `&CM` doorgeven. Lees het logbestand er maar eens op na. Als het eerste argument dat METAFONT ontvangt begint met een `&`, dan interpreteert METAFONT dat als de naam van een te laden base file. Men kan dus METAFONT naar zijn hand zetten door een eigen base file te laten laden.⁶

6 Modi

Een lettertype wordt altijd speciaal voor één uitvoerapparaat gemaakt, omdat elk uitvoerapparaat andere eigenschappen heeft, zoals het aantal pixels per inch, zwarting, vetheid van de pixels, etcetera. METAFONT kan daarmee allemaal rekening houden. Uiteindelijk doet METAFONT niets anders dan pixels zwart maken of wit laten, maar METAFONT kan op grond van die informatie wel beslissen of bijvoorbeeld dunne lijntjes het beste twee of drie pixels dik kunnen worden.

In het hiervoor genoemde bestand `atari.mf` staan voor vele uitvoerapparaten de bovengenoemde eigenschappen exact beschreven. Er wordt bijvoorbeeld een *modus atari* gedefinieerd, waar onder andere staat:

```
pixels_per_inch:=96; blacker:=0;
o_correction:=0.1
```

Omdat deze modi in de *plain base* zijn ingebakken, kan METAFONT direct aan het werk gezet worden voor elk in bestand `atari.mf` genoemd uitvoerapparaat.

⁶Zie METAFONT-boek blz. 35.

Dit gebeurt door aan het begin van een METAFONT-bestand een modus te vermelden of het als eerste argument mee te geven. `c_mf` doet dat laatste: als als *Device* de *HP Deskjet* is geselecteerd, geeft `c_mf` aan METAFONT mee: `mode=deskjet`; voor *Bildschirm* is dat `mode=atari`. Voor METAFONT heeft dit overigens pas effect als daarna opdracht `mode_setup` wordt gegeven. In kant-en-klare lettertypebestanden gebeurt dit al. Wordt geen modus gegeven, dan gaat METAFONT aan het werk voor `mode=proof`, waarover straks meer.

Het aanpassen van `atari.mf` is doorgaans alleen nodig als een nieuw uitvoerapparaat in gebruik wordt genomen dat niet (geheel) compatibel is met de erin genoemde apparaten. Een nuttige aanpassing is soms echter ook het wijzigen van de parameter *blacker*. Naarmate een afdrucker scherper afdruckt moeten dunne lijntjes iets meer gewicht krijgen. *blacker* moet dan worden vergroot. De in `atari.mf` genoemde factor 0,6 voor de Deskjet is echter aan de hoge kant. Ik heb er 0,2 van gemaakt, alle lettertype-bestanden weggegooid en in twee nachten nieuwe laten berekenen. Denk eraan dat voor lettertypes beginnend met *cm* de base `cm.bse` opnieuw moet worden aangemaakt.

7 Proefvelletjes

Tot nu toe kwam alleen het aanmaken van bestaande lettertypes aan bod, bijvoorbeeld voor een nieuw uitvoerapparaat of in een nog niet aanwezige vergroting. Men maakt dan gebruik van de noeste ontwerparbeid van Knuth en gebruikt zijn METAFONT-bestanden.

Het is ook mogelijk met METAFONT zelf nieuwe lettertypes of andere tekens te ontwerpen, of beter gezegd: op papier te ontwerpen en met METAFONT te maken. Bekijken we echter de omvang en complexiteit van Knuths METAFONT-bestanden, dan is het duidelijk dat enige bescheidenheid ons past. Niettemin: laat ons op verkenning gaan.

METAFONT biedt een fraaie mogelijkheid tot het bekijken en beoordelen van letterontwerpen: proefvelletjes. Dit is een speciale modus, `mode=proof`. In plaats van lettertekentjes van een millimeter of drie waaraan nauwelijks te zien is waarom precies ze nog niet naar wens zijn, zoals bij modi voor het scherm of een afdrucker, zorgt de proefmodus voor een soort proefvelletjes. Op elk vel wordt één teken flink vergroot weergegeven, in een kader dat de omvang aangeeft. De pixels worden weergegeven als grijze blokjes; gelabelde punten komen op de plaats van de referentiepunten waar doorheen METAFONT de curves trekt.

De gang van zaken voor het tonen van de proefvelletjes is geheel anders dan die voor het aanmaken van een lettertype. Op welk uitvoerapparaat de proefvelletjes uiteindelijk ook worden afgedrukt, het aantal pixels per inch is altijd hetzelfde: 2601,72. Dit is exact 36 pixels per punt. Een punt is een eenheid die veel gebruikt wordt in de wereld van druk en typografie als het om lettergroottes gaat; 1 punt is exact het 72,27-ste deel van een inch (25,4 mm), dus

ongeveer 0,35 millimeter. Het idee hierachter is dat uitvoerapparaten in het ontwerpstadium nog niet interessant zijn, de exacte verhoudingen, vorm en afmetingen van de lettertekens wel.

Elk pixel van het ontwerp wordt op het proefvelletje weergegeven door een grijs blokje van (standaard) 4 bij 4 pixels van het uitvoerapparaat. Een voorbeeld: moet een ontworpen teken 10 punten hoog worden, dan wordt het op het proefvelletje getekend in 10 maal 36 grijze blokjes hoog die elk een pixel voorstellen. De totale hoogte van het teken is dus 1440 pixels van het uitvoerapparaat en dat is op het scherm 15 inch en op de Deskjet 4,8 inch.

Voor het weergeven van de proefvelletjes is een speciaal programma aanwezig: `gftodvi.ttp`. METAFONT maakt van het lettertype-ontwerp in de proefmodus een bestand dat eindigt op `.gf`. Het programma `gftodvi` maakt hier rechtstreeks een `dvi`-bestand van. Dit programma draagt zorg voor het tekenen van het kader, het zetten van de titel, de labels bij de punten en het bepalen waar de grijze blokjes komen. Van T_EX wordt geen gebruik gemaakt. Hiervoor gebruikt `gftodvi` de bestanden `cmr8.tfm`, `cmtt10.tfm`, `gray.tfm` and `logo8.tfm`, die in de folder `tex\tfm` moeten staan. Het eerste en het laatste lettertype worden gebruikt voor het bovenschrift, het tweede voor de labels en `gray` voor de uitvergroete pixels.

Hierna kunnen we de proefvelletjes bekijken op bijvoorbeeld het scherm met het van T_EX bekende programma `dvi_view`. Dit programma heeft daarvoor de lettertypes `cmr8.pk`, `cmtt10.pk`, `gray.pk` and `logo8.pk` nodig, die zoals gebruikelijk in de folder `tex\fonts\font0096` moeten staan. Drie fonts hiervan kunnen op het scherm van `c_mf.prg` worden geselecteerd en zijn dus te maken zoals eerder beschreven. `Gray` ontbreekt echter.

De tekenset `gray`, die de grijze blokjes bevat, kan met Knuth's METAFONT-bestand `grayf.mf` gemaakt worden, te vinden in de folder `metafont\inputs\lib`. Dit kan het beste door een ander bestandje worden aangevoerd dat afgestemd is op het uitvoerapparaat zoals het scherm of de afdrucker.⁷ Bruikbaar voor het scherm is het volgende bestandje, dat we `gray.mf` noemen en in `metafont\inputs` zetten:

```
font_identifier "GRAY_ATARI";
input grayf
```

Klik op een leeg vakje in `c_mf` terwijl de control-toets wordt ingedrukt en tik `gray`. Nu kan `gray` zoals gebruikelijk worden aangemaakt. Nadat `gray`, `cmr8`, `cmtt10` en `logo8` voor het scherm op vergroting 1000 zijn aangemaakt, moeten de vier `.tfm`-bestanden nog in `tex\tfm` worden gezet.

8 Experimenteren met Metafont

Nu zijn alle bestanden aanwezig om met METAFONT te kunnen experimenteren. Het eenvoudigst is het het eerste experiment uit het METAFONTboek, dat op bladzijde 31, uit te voeren. Al direct zal een aantal zaken anders gaan.

⁷In appendix H van het METAFONTboek staan op blz. 332 METAFONT-bestandjes die `grayf` aanroepen.

Knuth meldt dat er iets op het scherm verschijnt, maar dat gebeurt op de ST niet. Christoph Strunk heeft op de Atari de opdracht `showit` niet geïmplementeerd.⁸ Wat de namen en de plaatsen van de bestanden betreft, biedt het bovenstaande meer houvast dan Knuth's boek.

Het is het overzichtelijkst om voor de experimenten een nieuwe folder te maken en hier alle invoer én uitvoer in te laten zetten. Maak bijvoorbeeld `metafont\exp` en maak hierin het bestandje `test.mf` met de tekst uit het METAFONTboek:

```
drawdot (35,70);
drawdot (65,70);
draw (20,40)..(50,25)..(80,40);
shipit; end.
```

Nu moet de hele machinerie in werking gezet worden en dat valt niet mee.

9 Gebruik `c_mf` in de proefmodus

De schil `c_mf` leent zich nauwelijks om experimenten met METAFONT in de proefmodus te doen.⁹ Na enig africhten lukt het echter wel. Start `c_mf`. Deselecteer alle lettertypes, klik op een leeg vakje terwijl de control-toets wordt ingedrukt, tik de naam van het testbestandje `test` en selecteer dit nieuwe vakje met een muisklik. Kies als hoofdfolder (onder *Finden*) `metafont\exp`. Wijzig *Environment* onder *Parameter* als volgt: verander `MFOUTPUTS` in een punt (dat is de hoofdfolder). Selecteer *.gf* *erhalten* onder *Parameter*: het `.gf`-bestand hebben we nodig. Kies als *externes Programm* `metafont\bin\gftodvi.ttp`. We hebben nog meer programmatuur nodig, maar daarvoor is binnen `c_mf` geen plaats. We zouden *Finden: GFtoPK* kunnen misbruiken door daar `gftodvi.ttp` selecteren, waardoor een ander *externes Programm* te gebruiken is. Dit heeft niet het gewenste effect, omdat `c_mf` er argumenten aan meegeeft die `gftodvi` niet begrijpt. Toch is het handig, omdat door de foutmelding voorkomen wordt dat `gftopk` het `.pk`-bestand aanmaakt. We hebben dit niet nodig, maar als het er is denkt `c_mf` dat het font er al is en wordt METAFONT niet gestart. We zouden `test.pk` dan elke keer zelf moeten weggooien.

De schil kent alleen echte devices, niet de proefmodus. Klik dus op het vak *Device*: om het met de hand te wijzigen. Verander *Name* in `proof`. Het *Font path* zetten we op `metafont\exp`. De resolutie doet er niet toe: het dient hier alleen om de naam van de folders met de lettertypes te bepalen. Maak er bijvoorbeeld 100 van. Wij gebruiken die folders niet, maar `c_mf` wil ze wel aanmaken en als `gftopk` wel wordt gedraaid, komt hier `test.pk` te staan. Zorg verder dat *Magnification* op 1000 staat.

Kies nu *Auswahl: Fehlende erzeugen* onder *Arbeiten* en METAFONT maakt van `test.mf` `test.gf`. We zien `gftodvi` een foutmelding opleveren en nemen dat voor

kennisgeving aan: druk op return. Start dan het externe programma `gftodvi` onder *Arbeiten*. Geef als parameter `\metafont \exp\test` in tegenstelling tot wat er wordt beweerd, blijkt het toch nodig het gehele pad op te geven. `gftodvi` maakt van `test.gf` `test.dvi`. De mogelijkheden van `c_mf` zijn nu uitgeput. Bewaar alle instellingen met *Speichern*!

Verlaat `c_mf` en start de schil rond T_EX: `ctex.prg`. Druk op H en kies als *Hauptdatei* `metafont\exp\test.mf`. T_EX kan hier niets mee, maar wij wel. Wijzig in *Environment* onder *Parameter* (met uw eigen driveletter):

```
TEXOUTPUTS = D:\METAFONT\EXP
```

`ctex` weet daar dan `test.dvi` te vinden. Bewaar de parameters. Gebeurt dat in `ctex.inf`, dan wordt het automatisch geladen bij het starten van `ctex.prg`: wel zo makkelijk.¹⁰ Druk nu op V, hetgeen `dvi_view` start en de test op het scherm zet. Ha!

De cyclus van een bestand wijzigen (toets E in `ctex`), door METAFONT halen, bewerken met `gftodvi` en bekijken met `dvi_view` is dus door omschakelen tussen `c_mf` en `ctex` uit te voeren. Door de parameters te bewaren wordt met het instellen daarvan geen tijd verloren. Echt prettig werkt het niet. Een volgende keer bekijken we het gebruik van een Unix-achtige schil hiervoor.

10 Enige principes van Metafont

Tot nu toe is METAFONT zelf nog niet aan de orde geweest. We zullen een tipje van de zware sluier oplichten door `test.mf` als volgt te wijzigen.

```
z1 = (35,70);
drawdot z1;
x2 = x1 + 30;
y2 = y1;
drawdot z2;
draw (20,40)..(50,25)..(80,40);
z3 = (50,0); z4 = (50,110);
draw z3..z4..cycle;
penlabels (1,2,3,4);
shipit;
end.
```

De eerste zes regels doen hetzelfde als de eerste drie regels van het vorige bestandje. Het is makkelijk met coördinaten te werken. Elk punt heeft een x-coördinaat (gerekend van links naar rechts) en een y-coördinaat (van onder naar boven). Punt 1 is aan te geven met `(x1,y1)`, maar in METAFONT is hiervoor ook de afkorting `z1` toegestaan. De eerste regel geeft tegelijk `x1` en `y1` een waarde. In de derde en vierde regel wordt punt 2 afgeleid uit punt 1. `y1=y2` betekent eenvoudig: zorg dat punt 1 en 2 op gelijke hoogte komen.

Alle coördinaten zijn uitgedrukt in pixels: daarmee werkt METAFONT. De derde regel betekent dat punt 2 30 pixels

⁸In de nieuwe versies van het T_EX-pakket — inmiddels MultiT_EX geheten — werkt dit wel. MultiT_EX is in Duitsland uitsluitend tegen betaling verkrijgbaar bij Maxon. Wellicht komen we hierop later terug.

⁹Bij MultiT_EX is met het gebruik van de proefmodus wel rekening gehouden.

¹⁰Het overschrijven van `ctex.inf` levert een write error op. Schrijf het eerst weg onder een andere naam, gooi op de desktop de oude `ctex.inf` weg en geef die naam aan het zojuist weggeschreven bestand.

verder naar rechts moet komen dan punt 1. In dit geval hebben die pixels geen relatie tot die van een uitvoerapparaat, omdat we in de proefmodus werken.

Met `draw z3...z4` tekent METAFONT een zo vloeiend mogelijke kromme door de punten 3 en 4. Kunst! Door twee punten is dat gewoon een rechte lijn! Afhankelijk van de coördinaten van 3, 4 en 5 zijn er echter heel fraaie krommen mogelijk met `draw z3...z4...z5`. Door hieraan `...cycle` toe te voegen wordt METAFONT gedwongen de kromme vloeiend met het beginpunt te verbinden. Wij schreven `draw z3...z4...cycle`: slechts twee punten; METAFONT kan haast niet anders dan daar een cirkel

doorheen tekenen, vanwege de eis dat de punten vloeiend moeten worden verbonden.

Het is handig te zien waar de punten nu precies staan waardoorheen METAFONT getekend heeft. Dat kan met `pen-labels` met als argument de punten die getoond moeten worden.

Voorlopig voldoende om mee te experimenteren! Een volgende keer komt het gebruik van een Unix-schil ter sprake, alsmede het afdrukken van proefvelletjes op een printer en de mogelijkheden daarvan.