

COMPONEREN MET DE COMPUTER

door L.G.L.T. Meertens

Voordracht gehouden voor het Nederlands Rekenmachine Genootschap op 25 september 1968

0. Inleiding

Computers kunnen gebruikt worden om muziek te componeren. Deze niet alledaagse toepassing is een antwoord op de uitdaging een creatief proces met behulp van een computer te realiseren, een uitdaging die waarschijnlijk zo oud is als de computer. Profiterend van de mogelijkheden die een computer biedt kan een componist iets tot stand brengen dat anders wegens de grote hoeveelheid daaraan verbonden werk niet gerealiseerd zou kunnen worden.

Dit is echter niet het enige doel dat met componeren per computer nagestreefd wordt. Als een theorie met betrekking tot muziek in een bepaalde stijl voldoende geformaliseerd is, kan men die theorie in een programma „vertalen” en vervolgens de door het programma gecomponeerde stukjes met de oorspronkelijke muziek in die stijl vergelijken. Op deze wijze valt vast te stellen in welke mate die theorie in staat is de schoonheid en inventiviteit te verklaren, die wij in de oorspronkelijke muziek ervaren, en hoeveel er nog te verklaren overblijft.

De meeste theorieën op muziekgebied lenen zich helaas slecht tot formalisering. Over een muzikaal zo belangrijk onderwerp als „melodie” is weinig meer te vinden, dan dat kleinere intervallen, met uitzondering van de relatief zeldzame priem, vaker voorkomen dan grotere, dat een groot interval bij voorkeur stapsgewijze (in de andere richting) weer wordt opgevuld, of dat een stijging en daling van de melodische lijn spanning verwekken en weer oplossen.

In een programma dat in een traditionele stijl componeert moet dan ook een groot deel nieuw gevormde theorie worden geïncorporeerd. Hoe zo'n programma in ALGOL 60 gerealiseerd is, zal in dit artikel worden geschetst.

1. Harmonie

De harmonieeler is van de onderdelen van de muziekler nog wel het meest vatbaar voor formalisering. Voor het hier beschreven programma is gebruik gemaakt van een boek van Hindemith, „Aufgaben für Harmonieschüler” [1]. Hierin worden op duidelijke wijze regels gegeven voor de opeenvolging van twee vierstemmige accoorden.

Enkele van deze regels zijn constructief, het merendeel daarentegen is in de vorm van een verbod gegoten. Een constructieve regel is: los een verminderde kwint of overmatige kwart in een accoord (bijv. in een dominant-septiem-accoord) op door met de ene stem bij het volgende accoord een seconde naar boven en met de andere een seconde naar beneden te gaan, zo dat een terts of een sext gevormd wordt. In figuur 1 wordt een overmatige kwart tussen de sopraan en de tenor opgelost.

	eerste accoord	tweede accoord
sopraan:	<i>fis</i>	<i>g</i>
alt:	<i>d</i>	<i>d</i>
tenor:	<i>c</i>	<i>b</i>
bas:	<i>d</i>	<i>g</i>

fig. 1. Oplossen van een overmatige kwart tussen sopraan en tenor

Een verbodsregel is: twee stemmen in een accoord die een kwint vormen, mogen niet door in dezelfde richting te *veranderen* in het volgende accoord weer een kwint vormen.

In figuur 2 komt tussen de sopraan en de bas een „kwintenparallel” voor.

	eerste accoord	tweede accoord
sopraan:	<i>c</i>	<i>d</i>
alt:	<i>f</i>	<i>d</i>
tenor:	<i>a</i>	<i>b</i>
bas:	<i>f</i>	<i>g</i>

fig. 2. Kwintenparallel tussen sopraan en bas

Bij het omzetten in programmavorm van deze regel waren twee moeilijkheden op te lossen. De eerste was dat in het genoemde boek van Hindemith soms, ongetwijfeld met reden, regels worden ingevoerd, om enkele bladzijden verder plots te worden opgeheven. De tweede was, dat niet bij voorbaat duidelijk is dat een reeks accoorden, volgens de regelen der kunst voortschrijdend, niet ooit een doodlopende weg zou kunnen inslaan en tenslotte een accoord zou kunnen bereiken vanwaar geen legitieme voortzetting mogelijk is.

De eerste moeilijkheid is opgelost door de verbodsregels zo te interpreteren, dat sommige verboden zeer sterk gelden, en dat andere daarentegen veeleer aanduiden dat een bepaald verschijnsel weliswaar bij voorkeur vermeden dient, maar in gevallen waar het bezwaarlijk te omzeilen valt, desnoods mag voorkomen. Van ieder verbod is de sterkte uitgedrukt in een getal, de bijbehorende „boete”.

In het programma is het harmonische gedeelte vastgelegd in de procedure *harmonic adviser*. Een van de input-parameters van deze procedure is het laatste accoord.

Harmonic adviser genereert, volgens constructieve principes, als mogelijke voortzetting van het laatste accoord een groot aantal nieuwe accoorden. Ieder van deze accoorden wordt achtereenvolgens getoetst aan alle in de

procedure opgenomen verbodsregels. Overtreedt het accoord een verbod, dan krijgt het de bijbehorende boete opgelegd. Per accoord worden de boeten opgeteld; het accoord met de laagste totale boete wordt afgeleverd in een output-parameter. Deze werkwijze biedt ook voor de tweede moeilijkheid een oplossing: in een moeilijk geval wordt toch nog de minst slechte voortzetting gekozen.

Het te vormen accoord kan van te voren reeds gedeeltematig bepaald zijn, doordat enkele stemmen al zijn ingevuld, of doordat de priem van het accoord gegeven is. Bovendien is bereikt, door gebruik te maken van het *Jensen device*, dat per aanroep van *harmonic adviser* met een willekeurig ingewikkelde wens rekening kan worden gehouden. Dit gebeurt door telkens wanneer een nieuw accoord gegenereerd is, het vast in de output-parameter in te vullen, en dan het resultaat van de evaluatie van een parameter *special fine* bij de totale boete op te tellen. Zo wordt desgewenst verkregen dat de sopraan en de bas in tegenbeweging zijn, of dat de sopraan en de alt in parallelle tertsen en sexten bewegen.

Enkele voorbeelden van verboden met hun boete zijn:

verbod	boete
kwintenparallel	100 000
verdubbelde terts	30 000
bas is niet verdubbeld	5 000
twee stemmen kruisen	2 000
verandering van de middenstemmen	100 à 200

voor iedere seconde verschil

Deze laatste verbodsregel vervangt de constructieve regel dat de middenstemmen (alt en tenor) over zo klein mogelijke intervallen bewegen, aangezien een grotere sprong in de middenstemmen verre verkieslijk is boven een kwintenparallel.

Om tijd te besparen worden in het programma steeds tevoren de sopraan en de bas ingevuld; dit besnoeit het aantal gegenereerde accoorden drastisch.

2. Melodie

De in deze paragraaf gegeven componeer-principes zijn alle, voorzover bekend, oorspronkelijk.

Voor het componeren van een passage wordt een recursieve procedure gebruikt, die zeer schetsmatig als volgt kan worden beschreven:

```

procedure componeer (duur, parameters);
if verdere onderverdeling ongewenst then
begin bereken linker duur;
        bereken linker parameters;
        componeer (linker duur, linker parameters);
        rechter duur: = duur — linker duur;
        bereken rechter parameters;
        componeer (rechter duur, rechter parameters)
end
  
```

end

Op een dergelijke wijze worden ritme en melodie gegenereerd.

2.1. Melodische lijn

Het principe dat ten grondslag ligt aan de vorming van de melodische lijn, vloeit voort uit de volgende, zeer algemene esthetische regel:

In een muziekstuk dient een geregelde afwisseling te zijn van spanning en ontspanning.

Veel componeer-principes kunnen in termen van span-

ning en ontspanning geformuleerd worden, en vallen dan onder deze regel. Bij de melodische lijn wordt spanning opgeroepen door een climax, en volgt bij daling weer ontspanning. Het werkt echter uitgesproken triviale, wanneer opeenvolgende hoogtepunten telkens op dezelfde toon vallen. Beter is het, wanneer ze iest verschillen. Dit kan heel behoorlijk als volgt bereikt worden:

Iedere passage bereikt ergens zijn hoogste toon. Wanneer deze passage in tweeën wordt verdeeld, heeft ook ieder van deze deelpassages een hoogste toon, en bij tenminste een van die twee deelpassages moet deze gelijk zijn aan de hoogste van de gehele passage. Het volstaat nu ervoor te zorgen dat de hoogste toon van de andere deelpassage iets lager is. Hetzelfde proces dat op de gehele passage is toegepast, wordt ook toegepast op de twee deelpassages.

In ALGOL 60:

```

procedure componeer (duur, hoogste toon);
value duur, hoogste toon; integer duur, hoogste toon;
if duur = 1 then output (hoogste toon) else
if RANDOM < 1/2 then
begin componeer (duur/2, hoogste toon);
        componeer (duur/2, hoogste toon — 1)
end else
begin componeer (duur/2, hoogste toon — 1);
        componeer (duur/2, hoogste toon)
  
```

end

De hierin voorkomende uitdrukking $RANDOM < 1/2$ behoeft enige verklaring. $RANDOM$ is een functie-procedure die bij iedere aanroep een ander reëel getal r oplevert, met $0 \leq r < 1$, en wel op zo'n manier dat, voor $0 \leq p \leq 1$, de uitdrukking $RANDOM < p$ een „kans” van nagenoeg p heeft waar te zijn. Bovendien valt er in opeenvolgende evaluaties van $RANDOM < p$ geen andere regelmaat te ontdekken dan al volgt uit de grootte p van de kans op de uitkomst „waar”.

$RANDOM$ kunnen we dus als een soort dobbelsteen gebruiken. Inplaats van echt te dobbelen, en te vragen: *gegooide ogen* = 6, kunnen we vragen: $RANDOM < 1/6$. In melodische lijnen die met bovenstaande procedure zijn gegenereerd, zal dus het hoogtepunt van de passage ongeveer even vaak in de linker- als in de rechterhelft van de passage vallen.

Een voorbeeld van een melodische lijn, die gegenereerd zou kunnen zijn door de aanroep *componeer* (16, 4):

1 0 1 2 1 2 3 2 1 2 3 2 3 4 3 2

oftewel, met de namen van de tonen:

d c d e d e f e d e f e f g f e

Louter toevallig zijn in dit voorbeeld alle intervallen een seconde.

De aanroep *componeer* (16, 4) resulteerde in dit voorbeeld in de aanroepen *componeer* (8, 3) en *componeer* (8, 4), en deze resulteerden weer in *componeer* (4, 2), *componeer* (4, 3), *componeer* (4, 3) en *componeer* (4, 4), enz.

Het volledige proces is aangegeven in figuur 3.

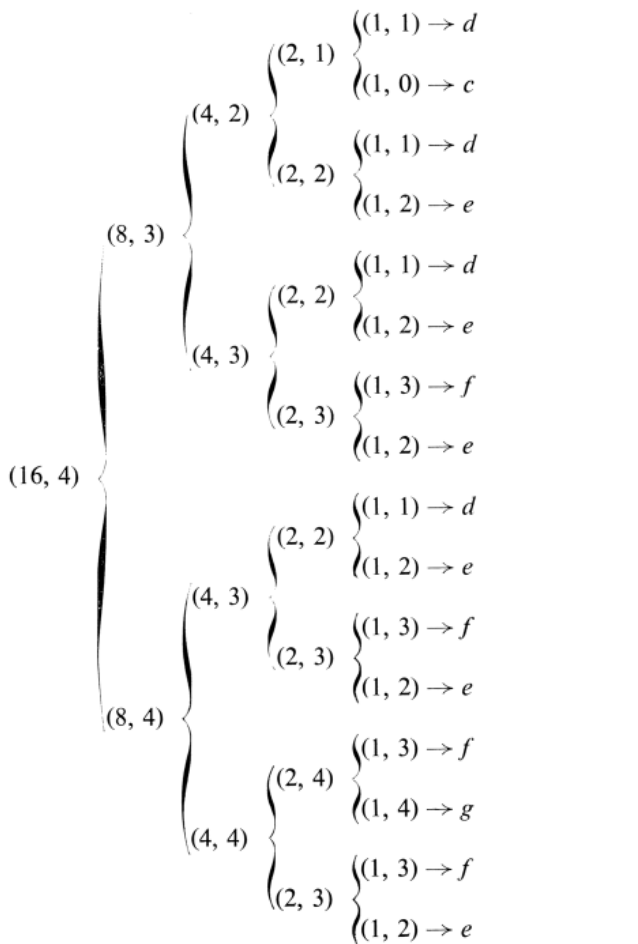


fig. 3. Voorbeeld van het genereren van een melodische lijn

Het is aardig een vergelijking te maken tussen frekwentie van voorkomen van de intervallen in de met dit principe gegenereerde melodische lijnen, en de intervallen uit klassieke muziek. In figuur 4 staan de geturfde frekwenties voor een gegenereerde reeks van 128 tonen.

interval	aantal malen voorgekomen
priem	25
seconde	83
ters	14
kwart	5
≥ kwint	0

fig. 4. Frekwentie van intervallen in gegenereerde reeks

Dit blijkt zeer goed in overeenstemming met het in de inleiding vermelde. Bovendien werden van de 19 voorkomende sprongen er 17 gevolgd door een seconde in de andere richting. Het blijkt dus dat dit eenvoudige principe, gebaseerd op een esthetische overweging, en passant een verklaring geeft voor twee bekende, schijnbaar onafhankelijke verschijnselen.

2.2 Ritme

Het ritme kan door de volgende procedure gegenereerd worden:

```

procedure componeer (duur, niveau);
value duur, niveau; real duur; integer niveau;
if RANDOM ≥ niveau/4 then output (duur) else
begin componeer (duur/2, niveau - 1);
      componeer (duur/2, niveau - 1)
end
  
```

We hebben in deze procedure, afhankelijk van *niveau*, de volgende kansen op onderverdeling:

niveau	duur	kans op onderverdeling
≥ 4	> 1	1
3	1	3/4
2	1/2	1/2
1	1/4	1/4
0	1/8	0

Een voorbeeld van een ritme dat gegenereerd zou kunnen zijn door de aanroep *componeer* (8, 6):

1 | 1/2 1/8 1/8 1/4 | 1/2 1/8 1/8 1/4 | 1 | 1/8 1/8 1/4 1/2 | 1/4 1/4 1/4 1/4 | 1/2 1/2 | 1

2.3. Synthese van ritme en melodische lijn

De procedures voor melodische lijn en ritme kunnen zeer gemakkelijk tot één procedure voor de melodie gecombineerd worden:

```

procedure componeer (duur, niveau, hoogste toon);
value duur, niveau, hoogste toon;
real duur; integer niveau; hoogste toon;
if RANDOM ≥ niveau/4 then
  output (hoogste toon, duur) else
if RANDOM < 1/2 then
begin componeer (duur/2, niveau - 1, hoogste toon);
      componeer (duur/2, niveau - 1, hoogste toon - 1)
end else
begin componeer (duur/2, niveau - 1, hoogste toon - 1);
      componeer (duur/2, niveau - 1, hoogste toon)
end
  
```

In het programma is deze procedure, daar *compose* geheten, veel ingewikkelder dan bovenstaande tekst doet vermoeden. Zo gebruikt de procedure 28 parameters, waarvan er 17 intensief meespelen in het recursieproces. Omwille van de duidelijkheid, en bovendien omdat de gehele tekst van de procedure enkele bladzijden zou beslaan, zijn hier slechts de eenvoudigste aspecten aangeroerd.

Vermeld mag nog worden dat de procedure, door met de reeks van *RANDOM*-waarden te manipuleren, het klaarspeelt in verschillende maten een gelijke reeks beslissingen in het splitsingsproces voor het ritme te nemen, zodat het effect ontstaat van ritmische motieven. Een poging met de melodie iets soortgelijks te doen, is door een programmeerfout mislukt.

De wijze waarop in de procedure *compose* een synthese ontstaat tussen harmonie en melodie is nog onbevredigend: de harmonische functie en de melodie worden onafhankelijk bepaald, en bij incompatibiliteit wordt de melodie iets gewijzigd.



fig. 5. Eerste zin van het Allegro

3. Macrostructuur

De macrostructuur van de compositie is die van een klassiek strijkkwartet. Hoewel er geen principiële moeilijkheid is per geval variaties toe te laten in de macrostructuur, is in het programma de volgende opeenvolging van vier delen vastgelegd:

- deel I *Allegro* $\frac{3}{4}$ -maat C groot
- deel II *Adagio* $\frac{6}{8}$ -maat a klein
- deel III *Menuet* $\frac{3}{4}$ -maat C groot
- deel IV *Finale* $\frac{2}{2}$ -maat C groot

Deel I wordt gecomponeerd in de sonate-vorm, bestaande uit expositie (herhaald), doorwerking en reprise. Ten gevolge van het drastische karakter van de gebruikte variatietechniek valt in de doorwerking bitter weinig te bespeuren van het oorspronkelijk thematisch materiaal. Het *menuet* is in een vorm, die bijv. bij Haydn vaak voorkomt. De macrostructuur van de compositie wordt tot stand gebracht door de procedure *compose* een aantal malen aan te roepen met geschikt gekozen parameters. Het is frappant hoe hiermee stukken van zeer verschillend karakter kunnen worden gecomponeerd.

4. Resultaat

Het hier beschreven programma heeft gedraaid op de ELECTROLOGICA X8 van het Mathematisch Centrum, en heeft de partituur voor een strijkkwartet geproduceerd. Het totale proces, van het ontwerp van het programma tot het eindresultaat heeft zich, in grote haast, binnen een maand afgespeeld. Het is dan ook niet mogelijk gebleken alle programmeerfouten in het programma op te sporen. In feite is het gecomponeerde strijkkwartet het resultaat van de eerste run zonder voortijdige beëindiging van het programma tengevolge van fouten; tevens tot nu de laatste run. Het werd ingezonden voor de IFIP computermuziek-wedstrijd 1968 en verkreeg daarin een bijzondere vermelding. In figuur 5, tot slot, een voorbeeld uit het strijkkwartet, gecomponeerd en getekend per computer.

Literatuur

- [1] P. Hindemith, *Aufgaben für Harmonieschüler*.
B. Schott's Söhne, Mainz.
Edition 3602.

Overdruk uit *Informatie* no. 12 - 1968