STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM
REKENAFDELING

MR 67

Automatisch tellen en scheiden van Nederlandse lettergrepen

door

H. Brandt Corstius

derde druk
maart 1967

The Mathematical Centre at Amsterdam, founded the 11th of February, 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications, and is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.) and the Central Organization for Applied Scientific Research in the Netherlands (T.N.O.), by the Municipality of Amsterdam and by several industries.

INHOUD

1. Inleiding

Het feit dat de meeste Nederlanders de meeste Nederlandse woorden op dezelfde manier in lettergrepen verdelen vormt een uitdaging om een rekenmachine te leren dat te doen. De officiele regels voor de verdeling van Nederlandse woorden in lettergrepen zoals die in [1] worden gegeven zijn niet direct bruikbaar. Ze hanteren begrippen als "uitspreekbaar" en "woordafleiding" die niet zonder meer voor een computer te gebruiken zijn.

In het algemeen kan men bij de scheiding in lettergrepen twee - vaak tegenstrijdige - principes onderscheiden:

1. Voeg het maximaal uitspreekbare bij de volgende lettergreep: AVON-TUREN

2. Scheid naar woordopbouw: AVOND-UREN

Voor woorden als BUURTJE,KWARTSLAGEN,LOODSPET is de betekenis,dus de context,onontbeerlijk. Hun aantal is zo klein dat we verder zullen aannemen dat we bij de identificatie van de lettergrepen alleen de letters van het woord zelf nodig hebben.

Een - triviale - oplossing van het probleem van de automatische lettergreepscheiding is om in het geheugen van de computer alle Nederlandse woorden met hun correcte splitsing op te nemen. Nog afgezien van de blijvende moeilijkheid in het Nederlands met de onvoorspelbare samengestelde woorden verwerpen we deze oplossing, omdat, zoals uit het vervolg blijkt, een oplossing die aanzienlijk minder geheugenruimte eist mogelijk is.

De onderzoekingen over hetzelfde probleem in het Engels[2],Frans[3],en Zweeds[4] hebben ons niet kunnen helpen. In de praktijk heeft men voor het Engels vaak de oplossing met het complete woordenboek gekozen [5].

Het gaat ons hier uitsluitend om de spellingslettergreep,dus niet de fonetische of fonologische lettergreep. Ook vragen we ons niet af of de bestaande spellingsregels voor lettergrepen zinvol zijn. We willen slechts de gebruikelijke manier van lettergreepscheiding door een rekenautomaat laten imiteren.

Eenvoudiger dan het splitsen in lettergrepen is het bepalen van het aantal lettergrepen in een Nederlands woord. Dit probleem wordt in paragraaf 2 opgelost. In paragraaf 3 wordt de scheiding van lettergrepen behandeld. We geven van beide problemen: de analyse,de oplossing in de vorm van een programma in ALGOL 60 [6],de behaalde resultaten,en enkele toepassingen.

2. Tellen van lettergrepen

2.1 Analyse

Gegeven een Nederlands woord in de een of andere voor een computer leesbare vorm. Gevraagd het aantal lettergrepen van dat woord.

We definieren:

e e n k l i n k e r: a,e,i,o,u,y

e e n m e d e k l i n k e r: b,c,d,...,x,z

e e n l e t t e r: eenklinker of eenmedeklinker

t w e e k l i n k e r: aa,au,ay,....,ij,uy

t w e e m e d e k l i n k e r: qu,ch

t w e e l e t t e r: tweeklinker of tweemedeklinker

d r i e k l i n k e r: aai,aau,aay,....eui,oei

k l i n k e r: eenklinker,tweeklinker of drieklinker

m e d e k l i n k e r: eenmedeklinker of tweemedeklinker

l e t t e r: klinker of medeklinker

o n v e r z a d i g d e l e t t e r: letter die als begin van tweeletter of drieklinker optreedt, b.v. a,q,aa,ie

v e r z a d i g e n d e l e t t e r: eenletter die als laatste element van tweeletter of drieklinker optreedt, b.v. a,i,j

Het te behandelen woord wordt in eerste instantie opgevat als een opeenvolging van eenletters. Deze worden vervolgens van links naar rechts tot maximaal grote letters bijeengevoegd. Deze comprimatie is eenvoudig te representeren in een matrix,waar elke rij hoort bij een onverzadigde letter en elke kolom bij een verzadigende letter. In de matrix staat de ontstane tweeletter of drieletter indien comprimatie mogelijk is. Van links naar rechts worden de onverzadigde letters van een woord zo mogelijk met hun verzadigende opvolger gecomprimeerd. Zo worden b.v. de vijf eenletters van AAIEN na comprimatie de drie letters AAI,E en N. De I kan niet met de E gecombineerd worden,omdat hij al als verzadigende letter bij de tweeklinker AA optrad. De - toevallige - omstandigheid dat de eerste twee letters van elke drieklinker steeds een tweeklinker vormen,maakt dit procede mogelijk. Dit geldt alleen niet voor de drieklinker EAU waarvoor dus speciale maatregelen getroffen moeten worden. Hiertoe is in de (E,A)-positie de waarde -1 geplaatst. De comprimatie-matrix is:

	A	E	I	U	O	Y	J	H
AA			AAI	AAU		AAY		
EE				EEU				
OO			OOI			OOY		
A	AA	AE	AI	AU		AY		
E	-1	EE	EI	EU		EY		
I		IE					IJ	
U			UI	UU		UY		
O		OE	OI	OU	OO	OY		
EU			EUI					
IE				IEU				
OE			OEI	OEU		OEY		
OU			OUI					
Q				QU				
C								CH

Voor het gebruik door de machine moeten de letters codegetallen krijgen. Hoe deze code wordt gekozen is wel niet essentieel,maar een aan het probleem aangepaste codering verhoogt de overzichtelijkheid en snelheid van het proces. Onze code is:

1	aai	17	ij	33	o	49	m
2	aay	18	oi	34	eu	50	n
3	oei	19	oui	35	ie	51	p
4	oey	20	oy	36	oe	52	r
5	ooi	21	ui	37	ou	53	s
6	ooy	22	uy	38	y	54	t
7	aau	23	eau	39	q	55	v
8	ae	24	oeu	40	c	56	w
9	ai	25	uu	41	j	57	x
10	au	26	aa	42	h	58	z
11	ay	27	ee	43	k	59	onderstrepingstrema
12	eeu	28	oo	44	b	60	qu
13	ei	29	a	45	d	61	ch
14	eui	30	e	46	f	62	apostrofe
15	ey	31	i	47	g	63	casedefinities
16	ieu	32	u	48	l	64	koppelteken

65 alle nietgenoemde tekens (leestekens etc.)
66 stopteken (aan eind van tekst)
68 symbolen die overgeslagen worden (tape feed,erase)

Door deze code kunnen de groepen als "klinkers","lettergreepscheiders","verzadigende letters" e.d. gemakkelijk onderscheiden worden. Dat de "Y" bij de klinkers wordt gerekend geeft in woorden als ROYAAL geen fouten, daar OY enz. tot klinkers worden gecomprimeerd. Voor het tellen van lettergrepen zou een eenvoudiger code mogelijk zijn,maar we willen dezelfde code gebruiken bij de lettergreepscheiding.

De met deze code in getallen vertaalde comprimatiematrix krijgt nullen op de lege plaatsen. Wanneer men in de comprimatie wijzigingen wil aanbrengen (b.v. ea als tweeklinker erkennen,of de ch niet comprimeren) dan kan dat zonder dat het programma gewijzigd hoeft te worden.

Nadat de eenletters van het te behandelen woord van links naar rechts gecomprimeerd zijn geldt: Het aantal lettergrepen is gelijk aan het aantal (gecomprimeerde) klinkers.

De enige benodigde informatie is dus: het klinker-zijn van bepaalde letters, en de comprimatiematrix.

Voorbeeld: QUADRAAT bestaat uit 8 eenletters. Na comprimatie zijn er vier medeklinkers: QU,D,R en T en twee klinkers : A en AA. Dus is het aantal lettergrepen twee.

2.2 Algolprogramma SYLTEL

Om het programma onafhankelijk te maken van de toevallig bij onze installatie gebruikte codering van de letters op de ponsband, zijn in het programma een aantal procedures opgenomen,waarvan de inhoud bij lezing kan worden overgeslagen. Het effect van deze procedures is het volgende:

De invoerprocedure "lees" krijgt de waarde in de boven beschreven codering van het eerstvolgende aangeboden symbool. De uitvoerprocedure "schrijf(aantal)" geeft de waarde van de variabele "aantal" aan een uitvoermechanisme. Voor het invoeren van getallen dient de invoerprocedure "read". De procedure "stop" stopt de machine; na het indrukken van een knop wordt het programma bij de volgende statement voortgezet.

Op de getallenband staan het array vertaal dat de codering van de symbolen verzorgt,en en het array comprimatie.

```
begin integer k,a,u,ea;
 integer array vertaal[0:127];
 integer procedure lees; begin LEES: k:=vertaal[RE7BIT];
   if k=66 then stop; if k=68∨k=63 then goto LEES; lees:=k end;
 procedure schrijf(aantal); integer aantal;
   begin NLCR;ABSFIXT(2,0,aantal) end;
 for k:=0 step 1 until 127 do vertaal[k]:=read;
 a:=29; u:=32; ea:=-1;

 begin integer i,j,letter,volgende letter,aantal lettergrepen;
  integer array comprimatie[26:40,29:42];
  boolean procedure echte letter(teken); integer teken;
    echte letter:= teken >0 ∧ teken < 63;
  boolean procedure klinker(letter); integer letter;
    klinker:= letter < 40;
  boolean procedure onverzadigd(letter); integer letter;
    onverzadigd:= letter > 25 ∧ letter < 41;
  boolean procedure verzadigend(letter); integer letter;
    verzadigend:= letter >  28 ∧ letter < 43;
  for i:=26 step 1 until 40 do for j:=29 step 1 until 42 do
  comprimatie[i,j]:= read; aantal lettergrepen:= 0;

LETTER: letter:=lees;
START: if echte letter(letter) then
  begin if klinker(letter) then
   begin if ¬onverzadigd(letter) then
    begin aantal lettergrepen:=aantal lettergrepen+1; goto LETTER
    end verzadigde klinker en dus lettergreep gevonden else
    begin volgende letter:= lees;
COMPRIMATIE: if verzadigend(volgende letter) then
     begin letter:=comprimatie[letter,volgende letter]; if letter=ea then
      begin aantal lettergrepen:= aantal lettergrepen+1; volgende letter:=
       lees; if volgende letter=u then goto LETTER else
       begin letter:=a; goto COMPRIMATIE
       end op ea volgde geen u
      end eau-geval; if echte letter(letter) then goto START
     end comprimatie; aantal lettergrepen:= aantal lettergrepen+1;
     letter:= volgende letter; goto START
    end onverzadigde klinker behandeld
   end klinker behandeld
  end echte letter behandeld else
  begin if aantal lettergrepen≠0 then
   begin schrijf(aantal lettergrepen); aantal lettergrepen:=0
   end aantal lettergrepen gevonden
  end woordscheider behandeld; goto LETTER
 end SYLTEL

end
```

Percentages van "lopende" krantenwoorden met bepaalde woordlengte en bepaald lettergreepaantal

lettergreepaantal / woordlengte	0	1	2	3	4	5	6	7	8	9	totaal
1	.11	.14									.24
2	.47	18.94	.03								19.45
3	.15	21.65	.04								21.84
4	.02	9.25	1.93	.01							11.20
5		3.24	4.20	.08							7.52
6		.50	8.22	.64	.01						9.36
7		.12	4.73	1.71	.15						6.71
8			1.70	3.73	.31						5.75
9			.74	3.37	.98	.02					5.11
10			.24	1.99	1.35	.11					3.69
11			.07	.95	1.74	.33	.01				3.10
12			.05	.39	1.15	.37	.02				1.98
13			.00	.14	.68	.32	.05				1.19
14				.04	.37	.39	.07	.06			.94
15				.01	.14	.27	.14	.03			.59
16				.00	.06	.16	.16	.04			.42
17					.02	.07	.11	.03	.01		.23
18					.01	.06	.15	.04	.01	.01	.28
19						.02	.08	.05	.01	.00	.17
20						.00	.05	.05	.01	.00	.12
21						.02	.02	.05	.01		.10
totaal	.75	53.83	21.96	13.05	6.97	2.16	.86	.36	.05	.01	

Percentages van "verschillende" krantenwoorden met bepaalde woordlengte en bepaald lettergreepaantal

lettergreepaantal / woordlengte	0	1	2	3	4	5	6	7	8	9	totaal
1	.06	.06									.12
2	.48	1.69	.04								2.20
3	.23	4.76	.06								5.06
4	.04	7.08	1.59	.01							8.72
5		4.13	5.79	.12							10.04
6		.91	11.24	.97	.02						13.14
7		.18	8.45	3.31	.23						12.17
8			3.56	7.28	.53						11.37
9			1.46	6.69	1.88	.06					10.09
10			.48	4.34	2.83	.20					7.85
11			.12	2.14	3.63	.39	.02				6.31
12			.06	.78	2.36	.82	.04				4.06
13			.01	.30	1.46	.80	.12				2.68
14				.08	.77	.70	.15	.09			1.78
15				.02	.33	.61	.30	.04			1.30
16				.01	.14	.38	.34	.10			.96
17					.04	.18	.27	.07	.02		.58
18					.03	.14	.32	.11	.02	.02	.65
19						.06	.20	.12	.02	.01	.40
20						.01	.11	.13	.03	.01	.29
21						.03	.05	.12	.03		.23
totaal	.82	18.80	32.86	26.04	14.25	4.37	1.92	.78	.13	.03	

2.3 Resultaten

De gevonden fouten zijn in vier groepen te verdelen:

1. Vreemde woorden als PE-A-CE. Deze vallen buiten ons probleem.

2. De door ons gebruikte krantenteksten bevatten geen trema's,omdat de flexowriter die niet kent. Hierdoor werd het aantal lettergrepen van BELGIE twee. Een trema zal of als niet spatiegevend symbool voor de klinker worden gezet,of samen met de klinker een nieuw symbool vormen. In beide gevallen is de telling eenvoudig programmeerbaar. Ons programma telt juist indien men de trema in de vorm van een onderstreping aanbrengt: BELGIE krijgt 3 lettergrepen. Door de trema als medeklinker te rekenen gaat dan ook de scheiding in lettergrepen goed.

3. Abnormaliteiten als WTTEWAAL.

N.B.:afkortingen (als KLM) worden als nullettergrepige woorden beschouwd.

4. Echte fouten door onjuiste comprimatie: MUSEUM krijgt ten onrechte 2 lettergrepen.

De zeldzaamheid van woorden als WTTEWAAL en MUSEUM maakt dat we het probleem van de automatische bepaling van het aantal lettergrepen in een Nederlands woord als opgelost mogen beschouwen.

2.4. Toepassingen

2.4.1. Taalstatistiek

Met een enigszins gewijzigd programma werden de krantenwoorden uit [7] behandeld.Voor elke krant afzonderlijk werden de aantallen woorden geteld van een bepaalde woordlengte in eenletters (alleen de combinatie IJ werd gecomprimeerd tot een letter) en een bepaald lettergreepaantal. Dit werd gedaan voor de "lopende"woorden (elk woord zo vaak het voorkomt) en voor de "verschillende" woorden (elk woord maar een maal geteld).

De gevonden percentages voor 45004 lopende woorden (16192 verschillende woorden) zijn te vinden op pagina 6 en 7.

Het aantal lopende woorden met echte lettergrepen bedroeg 44666, het aantal lettergrepen daarin 82692 en het aantal letters 241344. Gemiddeld telt een lopend woord dus 1.85 lettergrepen en telt een lettergreep uit een lopend woord 2.92 letters.

Het aantal verschillende woorden met echte lettergrepen bedroeg 16060, het aantal lettergrepen daarin 42068 en het aantal letters 126261. Gemiddeld telt een "verschillend" woord dus 2.62 lettergrepen en telt een lettergreep uit een verschillend woord 3.00 letters. Het Nederlands neemt hiermee in de tabel van Fucks [8] een plaats in tussen Duits en Arabisch.

Eenlettergreepwoorden van 7 letters waren in onze tekst SPREEKT en SLECHTS (SCHRAALST is langer),het kortste negenlettergreepwoord was ANTIREVOLUTIONAIRE.

2.4.2 Poetische analyse

Hoewel de spellingslettergreep niet hetzelfde is als de poetische lettergreep werden met een enigszins gewijzigd programma de aantallen lettergrepen per versregel bepaald. En passant werd hierbij door de computer het rijmschema bepaald: hoofdletters voor mannelijk rijm, kleine letters voor vrouwelijk rijm. Een voorbeeld:

A.C.W. Staring, HET HONDENGEVECHT

	lettergreeptal	rijm
Bereisde Roel zag op zijn tochten	9	a
geweldig veel. Twee bullebijters vochten,	11	a
voor 't wijnhuis,in een kleine Poolse stad,	10	B
terwijl hij juist aan 't venster zat:	8	B
"Zulk vechten,mensen. - - Zij verslonden	9	c
malkander letterlijk. Met iedren hap,ging oor	12	D
of poot er af - en glad als vet er door.	10	D
Ons scheiden kwam te laat. wij vonden	9	c
het restjen: - op mijn eer,	6	E
de staarten, en niets meer."	6	E

2.5. Opmerkingen

Hetzelfde procede lijkt ook voor andere talen bruikbaar.

De "deuk" in het woordlengtehistogram bij de lengte 5 is verklaarbaar uit de overgang van 1 op 2 lettergrepen.

Als maat voor de lengte van een woord is het lettergreepaantal wezenlijker dan het letteraantal (het is b.v. invariant t.o.v. veranderingen in de spelling).

3. Scheiden van lettergrepen

3.1. Analyse

Gegeven een Nederlands woord in de een of andere voor een computer leesbare vorm. Gevraagd wordt om tussen de lettergrepen van dat woord koppeltekens aan te brengen.

De algemene gang van de oplossing is als volgt:

1. Vertaal de eenletters van het woord in bruikbaarder codering (die van 2.1).

2. Comprimeer de eenletters (zoals beschreven in 2.1).

3. Hak achtervoegsels af. Hierdoor krijgt men bijvoorbeeld WOEST-HEID i.p.v. WOES-THEID.

4. Zoek de eerste twee gecomprimeerde klinkers. Zijn er geen twee dan is proces afgelopen. Zijn ze er wel,dan moet ergens tussen die twee klinkers gesplitst worden volgens vier voorschriften "tegelijk":

4.1. Voorvoegsels afsplitsen als dat nuttig is: hierdoor krijgt men bijvoorbeeld ON-EENS i.p.v. O-NEENS (niet altijd ON afsplitsen,anders: ON-TZETTEND).

4.2. Letters als AA i.h.a. niet aan het eind van lettergreep: RAAM-ANTENNE en niet RAA-MANTENNE.

Letters als AAI i.h.a. aan het eind van lettergreep: LAWAAI-SFEER en niet LAWAAIS-FEER.

4.3. Tussen de twee eerste klinkers staan nu 0,1,2, of meer medeklinkers. We geven een klinker aan door "V",een medeklinker door "C" en een willekeurige letter door "X".

4.3.1. 0 medeklinkers,dan splitsing V-V. PAPOE-A

4.3.2. 1 medeklinker,dan splitsing V-CV. VOE-TEN

4.3.3. 2 medeklinkers.Als hun combinatie uitspreekbaar is,dan splitsing V-CCV. PRO-BLEEM anders splitsing VC-CV. VOER-DEN

4.3.4. meer dan twee medeklinkers.Als de laatste drie uitspreekbaar dan splitsing -CCCV. V.b.: ANGST-SCHREEUW Als alleen de laatste twee uitspreekbaar dan splitsing C-CCV. V.b.: ANGST-KREET Anders splitsing CC-CV. V.b.: ANGST-GIL

4.4. Regels voor speciale gevallen.

5. Als op deze manier de voorste lettergreep is afgesplitst wordt regel 4 geïtereerd.

De benodigde informatie bestaat dus uit:

1. Kennis van de letters,klinkers,lettergreepscheiders etc.

2. De comprimatieregels.

3. De uitspreekbare CC-combinaties. Hun verzameling noemen we t w e e c o n s.

4. De uitspreekbare CCC-combinaties. Hun verzameling noemen we d r i e c o n s.

5. De achtervoegsels.

6. De voorvoegsels met opgave in welke gevallen ze nuttig zijn.

De eerste twee groepen informatie zijn in 2.1 behandeld. De laatste vier groepen zijn sterk onderling afhankelijk. Elke verandering in de een heeft consequenties voor de andere.

De tweecons. Op grond van ons krantenmateriaal namen we in de tweecons op:

PL PR TR CHR

BL BR CL CR DR FL GR KL KR KW SC SCH TH VL VR ZW

FR ST SP

Niet opgenomen werden dus o.a. SF SJ SK SL SM SN KN KJ PN PJ TS DW TW WR. Deze zouden namelijk meer foute dan juiste splitsingen veroorzaken. De laatste drie (FR, ST en SP) worden alleen als element van tweecons beschouwd als ze door nog een medeklinker worden voorafgegaan. Tussen twee klinkers worden ze gesplitst in F-R, S-T en S-P.

De driecons . Hierin zijn opgenomen: SCHR SPL SPR STR. Ze bestaan dus alle vier uit een S en een van de eerste vier elementen van tweecons.

De rol van voorvoegsels en achtervoegsels is niet symmetrisch. Dit wordt veroorzaakt doordat we i.h.a. meer weten over het mogelijke begin dan over het mogelijke eind van een Nederlandse lettergreep,en omdat nuttige achtervoegsels i.h.a. inderdaad achteraan het woord staan,terwijl nuttige voorvoegsels overal in het woord kunnen voorkomen. Achtervoegsels worden dan ook in het begin van het proces van achteren afgesplitst, maar voorvoegsels worden bij elke splitsing betrokken. Lang niet alle conventionele voor- en achtervoegsels hebben een nuttig effect. We beperkten ons tot eenlettergrepige voor- en achtervoegsels van maximaal vier letters (hieronder hoort dus ook b.v. SCHIER).

De nuttige achtervoegsels zijn te verdelen in vier groepen:

1. Die waarvan de eerste letter een medeklinker is die als laatste letter in een element van tweecons of driecons voorkomt (ook als de eerste twee letters van het achtervoegsel aan het eind van een element van driecons staan zou het achtervoegsel nuttig zijn,maar dit soort hebben wij niet gebruikt).

Zo verhinderen de achtervoegsels HEID en WIJS de splitsingen WOES-THEID en JAZ-ZWIJS .

2. Achtervoegsels die beginnen met een CC-combinatie die niet in tweecons staat,maar wel voorkomt als uitspreekbare combinatie.

3. Achtervoegsels die met een klinker beginnen (deze kunnen ook snel verkeerde splitsingen veroorzaken).

Zo verhindert AARD de splitsing LA-FAARD.

4. Achtervoegsels die terwille van de snelheid van het programma worden opgenomen,maar geen verkeerde splitsingen voorkomen. Zo zou men kunnen opnemen ZAAM en SCHAP.

De nuttige voorvoegsels worden onderscheiden in vier types: VC,CVC, VCC, en XXCC. Hun gebruik is afhankelijk van de positie van de tweede gecomprimeerde klinker in het te splitsen woord. Als deze op de 3e,4e,5e of 6e plaats staat,worden de volgende splitsingen vermeden:

Tweede klinker op 3e plaats	4e plaats	5e plaats	6e plaats
Type voorvoegsel			
VC o-necht	ui-trit	ij-spret	
CVC	wa-norde	di-sponibel	di-splezier
VCC	al-sof	an-twoord	
XXCC		hoof-donderwijzer	aart-spiekeraar

Er zijn nog andere gevallen mogelijk (SCHIE-REILAND b.v.) maar die werden niet gebruikt.

Een geval op zichzelf vormen die voorvoegsels die altijd afgesplitst moeten worden,zoals HOF (waardoor HO-FAUTO voorkomen wordt):Een systematisch onderzoek hiernaar vond niet plaats.Ook kan men voorvoegsels invoeren die niet werken als ze voorafgaan aan de letter E.Hierdoor zouden de voorvoegsels RAND en FILM RAN-DAPPARATUUR en FIL-MACTEUR voorkomen,maar toch RAN-DEN en FIL-MEN blijven geven.

Afhankelijk van de positie van de tweede gecomprimeerde klinker wordt een bepaald gedeelte van de voorvoegsellijst afgezocht naar afsplitsbare voorvoegsels. Hierdoor is bijvoorbeeld het ON,ONT en het VOOR,VOORT probleem in veel gevallen op te lossen.

Wij gebruikten de volgende achtervoegsels:

AARD HEID LAAN LIJK LOOS WIJS REN RIJK LAND LING RING

en de volgende voorvoegsels:

AAN OOR IN ON OER ER AF AB

DAAR DER HIER VOOR VEN VER HER WAN MAAT SUB ZUID DIS

VAN NOG HOF JET

ALS ONT

AARTS VOORT HOOFD NOORD GROOT

Deze lijsten zijn zeker voor verbetering vatbaar. Onze op grond van successieve bewerkingen van krantenmateriaal aangebrachte wijzigingen bestonden voornamelijk uit het weglaten van voor- en achtervoegsels die teveel fouten veroorzaakten. De keuze wordt mede bepaald door de gekozen tweecons en driecons. De - vreemde - voorvoegsels DER,VEN,en VER werden genomen om de echte voorvoegsels ONDER,BOVEN,en OVER af te kunnen splitsen (we beperkten ons immers tot eenlettergrepige voorvoegsels). De fouten die DER veroorzaakt in woorden als STUDEREN worden opgeheven door REN als achtervoegsel op te nemen (achtervoegsels afsplitsen heeft prioriteit boven voorvoegsels afsplitsen).

Wijzigingen in deze lijsten zijn gemakkelijk aan te brengen daar ze in het programma zelf niet voorkomen.

De hier genoemde voorvoegsels en achtervoegsels zijn uitverkozen op grond van krantenmateriaal. De frekwentie van de woorden speelt hierin dus mee (Omdat ADENAUER meer voorkomt dan ADORATIE is AD niet bij de voorvoegsels gezet. Dit kan zich dus met de tijd wijzigen).

De informatie in tweecons,driecons,voorvoegsels en achtervoegsels wordt numeriek gemaakt door een opeenvolging van letters op te vatten als een getal in het 64-tallig stelsel. Zo krijgt b.v. SCHR de waarde :

code(S)×64×64 + code(CH)×64 + code(R) = 221044.

Tenslotte zijn er nog speciale maatregelen genomen. Het tot dusver beschreven systeem zou namelijk nog fouten maken als:(van iedere categorie wordt een voorbeeld gegeven) GEK-NEVELD,BED-WONGEN,BURE-AU,E-XA-MEN,MEEN-EMEN,ZEEM-AN, INDU-STRIE, PER-SCONFERENTIE, NATI-ON-AAL,REGERING-SONDERHANDELINGEN,TOES-TANDEN. Zij worden door speciale regels toch goed gesplitst. Van belang is hierbij alleen het begrip van de eindloze achtervoegsels en beginloze voorvoegsels: STAND wordt niet als achtervoegsel beschouwd,maar er wordt altijd voor afgebro-ken,evenzo wordt altijd tussen NGS en een klinker gesplitst.

Ad hoc-maatregelen die maar voor een enkel geval zouden gelden zijn niet getroffen.

Op een punt wijkt het aantal afgesplitste lettergrepen af van het aantal bepaald volgens paragraaf 2: Volgens [1] moet bij een "x" tussen twee klinkers niet gesplitst worden. EXAMEN heeft dus drie lettergrepen,maar wordt gescheiden als EXA-MEN.

Niet gerealiseerd werden splitsingen waarbij letters verdwijnen,zoals in STROOTJE,MENUUTJE en OPAATJE.

3.2. Algolprogramma SYLSPLITS

Weer werd getracht om het ALGOL-programma onafhankelijk te maken van de toevallig gebruikte installatie. Het array "vertaal" zorgt weer voor de codering van de letters. Het array "WOORD" bevat de letters in de installatie-afhankelijke code,die onbekend kan blijven omdat de letters van WOORD weer net zo uitgevoerd worden,met koppeltekens tussen de let-tergrepen. De in onze code vertaalde en later gecomprimeerde letters ko-men in het array "woord". Tussen woord en WOORD wordt een verwij-zingslijst bijgehouden. De invoerprocedure "lees" krijgt de waarde in de installatie-code van het eerstvolgende aangeboden symbool. De uitvoerpro-cedure "uitvoer(teken)" geeft de waarde van "teken" aan een uitvoerappa-raat. Het enige symbool waarvan we de installatiecode willen weten is het koppelteken. We nemen aan dat het koppelteken in de installatiecode dezelfde waarde heeft als in onze code,dus 64 (zowel in lower als in upper case).

"geen letter maar wel element van woord" zijn bijvoorbeeld de lower en de UPPER case. Het zijn geen lettergreepscheiders maar ook geen let-ters.

```
begin integer a,d,e,k,n,o,p,s,t,u,x,be,ge,sc,st,gs,au,eau,mee,wee,zee,mede
  klinker,koppelteken,stopteken,geen letter maar wel element van woord;
  integer array vertaal[0:127];
  integer procedure lees; begin integer a; LEES: a:=RE7BIT;
    if a=0∨a=127 then goto LEES; lees:=a end;
  procedure uitvoer(teken); integer teken; PU7BIT(teken);
  for k:=0 step 1 until 127 do vertaal[k]:=read;
  a:=29; d:=45; e:=30; k:=43; n:=50; o:=33; p:=51; s:=53; t:=54; u:=32; x:=57;
  be:=2846; ge:=3038; sc:=3432; st:=3446; gs:=3061; au:=10; eau:=23;
  mee:=3163; wee:=3611; zee:=3739; medeklinker:=59; stopteken:=66;
  geen letter maar wel element van woord:=63; koppelteken:=64;
  begin integer h,i,j,eerste klinker,tweede klinker,woordbegin,woordeind,
      WOORDbegin,WOORDeind,symbool,letter,volgende letter,aanvang,slot;
    integer array woord,WOORD,verwijzing van woord naar WOORD[1:50],
      comprimatie[26:40,29:42],voorvoegsel[2:4,0:18],achtervoegsel[3:4,1:8],
      tweecons[1:23];
    boolean procedure echte letter(teken); integer teken;
      echte letter:=teken<63;
    boolean procedure klinker(letter); integer letter;
      klinker:=letter<39;
    boolean procedure onverzadigd(letter); integer letter;
      onverzadigd:=letter>25 ∧ letter<41;
    boolean procedure verzadigend(letter); integer letter;
      verzadigend:=letter>28 ∧ letter<43;
    boolean procedure aa groep(letter); integer letter;
      aa groep:=letter<29 ∧ letter>24;
    boolean procedure aai groep(letter); integer letter;
      aai groep:=letter<6;
    procedure klinkervraag; begin boolean voorste klinker gevonden;
      voorste klinker gevonden:=false;
      for i:=woordbegin step 1 until woordeind do
      begin if klinker(woord[i]) then begin if voorste klinker gevonden then
       begin tweede klinker:=i; goto EXIT end else begin eerste klinker:=i;
       voorste klinker gevonden:=true end end end Als er geen twee
      klinkers meer in woord voorkomen wordt WOORD uitgevoerd:;
      for i:=WOORDbegin step 1 until WOORDeind do uitvoer(WOORD[i]);
    uitvoer(symbool); i:=j:=1; goto OPBOUW; EXIT: end klinkervraag;
```

```
    procedure splitsaf (h); integer h;
      begin woordbegin:=woordbegin+h; for i:=WOORDbegin step 1 until
      verwijzing van woord naar WOORD[woordbegin]-1
      do uitvoer(WOORD[i]); uitvoer(koppelteken); WOORDbegin:=verwijzing
      van woord naar WOORD[woordbegin]; goto REST end splitsaf;
    procedure voorvgl (h); value h; integer h; begin integer aanvang;
      aanvang:=woord[woordbegin]×64+woord[woordbegin+1];
      if aanvang=be ∨ aanvang=ge then begin letter:=woord[woordbegin+2];
      if woordeind>5 ∧ (letter=d ∨ letter=k ∨ letter=s ∨ letter=t) then
      splitsaf(2) else goto ERUIT end van de afsplitsing van be- en ge-;
      if h>4 ∨ h<2 then goto ERUIT;
      if h>2 then aanvang:=aanvang×64 + woord[woordbegin+2];
      if h>3 then aanvang:=aanvang×64 + woord[woordbegin+3];
      for j:=voorvoegsel[h,0] step 1 until 18 do begin if aanvang=
      voorvoegsel[h,j] then splitsaf(h) end; ERUIT: end voorvgl;
    for i:=26 step 1 until 40 do for j:=29 step 1 until 42 do
    comprimatie[i,j]:=read;
    for i:=2,3,4 do for j:=0 step 1 until 18 do voorvoegsel[i,j]:=read;
    for i:=3,4 do for j:=1 step 1 until 8 do achtervoegsel[i,j]:=read;
    for i:=1 step 1 until 23 do tweecons[i]:=read; i:=j:=1;
    comment nu wordt de te splitsen tekst ingelezen;

OPBOUW: symbool:=lees; letter:=vertaal[symbool]; if letter=stopteken then
  begin stop; i:=j:=1; goto OPBOUW end herstartbare stop;
  if echte letter(letter) then begin WOORD[i]:=symbool; woord[j]:=letter;
  verwijzing van woord naar WOORD[j]:=i; i:=i+1; j:=j+1; if i=51 then
  begin symbool:=lees; goto TEVEEL end te lang woord;
  goto OPBOUW end echte letter in woord en WOORD gezet;
  if i=1 then begin uitvoer(symbool); goto OPBOUW end uitvoer scheider;
  if letter=geen letter maar wel element van woord then begin WOORD[i]:=
  symbool; i:=i+1; goto OPBOUW end wel in WOORD,niet in woord gezet;
  if i=2 then begin uitvoer(WOORD[1]); uitvoer(symbool);
  i:=j:=1; goto OPBOUW end eenletterwoord hoeft niet gesplitst;
TEVEEL: WOORDeind:=i-1; woordeind:=j-1;
  WOORDbegin:=woordbegin:=i:=j:=1; h:=0; goto OPGEBOUWD;
```

```
COMPRIMATIE: if onverzadigd(letter) then
  begin volgende letter:=woord[i+1]; if verzadigend(volgende letter) then
    begin letter:=comprimatie[letter,volgende letter]; if letter>0 then
      begin i:=i+1; h:=h+1; woord[i]:=letter; goto OPGEBOUWD
  end end end comprimatie; woord[j]:=woord[i]; j:=j+1; i:=i+1; verwijzing
  van woord naar WOORD[j]:=verwijzing van woord naar WOORD[j]+h;
OPGEBOUWD: if i≠woordeind then begin letter:=woord[i];
  goto COMPRIMATIE end else
  begin woord[j]:=woord[woordeind]; woordeind:=j end woordcomprimatie;
GECOMPRIMEERD: klinkervraag; if woordeind>3 then begin  h:=3;
   slot:= woord[woordeind]+64×woord[woordeind-1]+4096×woord[woordeind-2];
ACHTERVGL: for i:= 1step 1 until 8 do
begin if slot=achtervoegsel[h,i] then
   begin woordeind:=woordeind-h; WOORDeind:=WOORDeind+1;
     for j:=WOORDeind step -1 until verwijzing van woord naar WOORD
     [woordeind+1]+1 do WOORD[j]:=WOORD[j-1];
     WOORD[verwijzing van woord naar WOORD[woordeind+1]]:=koppelteken;
     goto GECOMPRIMEERD end koppelteken gezet end achtervgl;
  if h=3 then begin h:=4; slot:=slot+ 262144×woord[woordeind-3];
   goto ACHTERVGL end achtervoegsels van vier letters end;
REST: klinkervraag; if aai groep(woord[eerste klinker]) then
  splitsaf(eerste klinker-woordbegin+1);
  if tweede klinker-eerste klinker=1 then goto KLINKERS;
  if tweede klinker-eerste klinker=2 then goto EEN MEDEKLINKER;
  GEVAL VAN MEER DAN EEN MEDEKLINKER:
  aanvang:=woord[tweede klinker-2]×64 + woord[tweede klinker -1];
  for i:=1 step 1 until 23 do begin if aanvang=tweecons[i] then
  begin if i<5 ∧ woord[tweede klinker-3]=s then
    begin if woord[tweede klinker-4]=u then splitsaf(tweede klinker-woordbegin
      -2); if woordeind>16 ∧ i=2 then begin if tweede klinker<woordeind-2 ∧
      tweede klinker>9 then splitsaf(tweede klinker-woordbegin-2) end s in
      samenstelling; voorvgl(tweede klinker-woordbegin-2);
      if aa groep(woord[eerste klinker]) ∧ tweede klinker-eerste klinker=4
      then splitsaf(eerste klinker-woordbegin+2) else splitsaf
      (tweede klinker-woordbegin-3) end driecons geval;
```

```
      if aa groep(woord[eerste klinker]) ∧ tweede klinker-eerste klinker=3
      then splitsaf (eerste klinker-woordbegin+2);
      voorvgl(tweede klinker-woordbegin-1);
      if i>20 then begin if eerste klinker=tweede klinker-3 ∧¬(aanvang=st ∧
      woord[tweede klinker]=a ∧ woord[tweede klinker+1]=n ∧ woord
      [tweede klinker+2]=d) then splitsaf(tweede klinker-woordbegin-1) end
      FR SP en ST tussen klinkers splitsen;
      if aanvang=sc then begin if  ¬ klinker(woord[tweede klinker-3]) then
      splitsaf(tweede klinker-woordbegin-1) end SC;
      if tweede klinker-eerste klinker=3 ∧ woord[eerste klinker]=o ∧ woord
      [eerste klinker+1]=p then splitsaf(2); splitsaf(tweede klinker-woordbegin-2)
    end end tweeconsgeval;
    if aanvang=gs then begin if woord[tweede klinker-3]=n then
    splitsaf(tweede klinker-woordbegin) end NGS-klinker;
    voorvgl(tweede klinker-woordbegin-1); voorvgl(tweedeklinker-woordbegin-2);
    voorvgl(tweede klinker- woordbegin); splitsaf(tweede klinker-woordbegin-1);

KLINKERS: if woord[eerste klinker]=e ∧ woord[tweede klinker]=au then
    begin woord[eerste klinker]:=eau; woordeind:=woordeind-1;
      for i:=eerste klinker+1 step 1 until woordeind do
      begin woord[i]:=woord[i+1]; verwijzingvanwoordnaarWOORD[i]:=verwijzing
      van woord naar WOORD[i+1] end; goto REST end EAU-comprimatie;
    splitsaf(eerste klinker-woordbegin+1);

EEN MEDEKLINKER: if woord[eerste klinker+1]=x then begin woord[eerste
    klinker]:=medeklinker; goto REST end x tussen 2 klinkers:geen splitsing;
    if eerste klinker>woordbegin ∧ aa groep(woord[eerste klinker]) then
    begin aanvang:=64×woord[eerste klinker-1]+woord[eerste klinker];
      if aanvang=mee∨aanvang=wee∨aanvang=zee then splitsaf(eerste klinker
      -woordbegin+1) else splitsaf (eerste klinker-woordbegin+2) end EE;
    if eerste klinker>1  then begin if klinker(woord[eerste klinker-1]) ∧
    woord[woordbegin]=o ∧ woord[woordbegin+1]=n then splitsaf(1) end;
    if tweede klinker<woordeind then voorvgl (tweede klinker-woordbegin);
    splitsaf(tweede klinker - woordbegin-1)
  end SYLSPLITS
end
```

3.3 Resultaten

Het programma SYLSPLITS werd toegepast op de 43712 woorden (4114 verschillende woorden) uit [7]. Er zijn vele methoden mogelijk om het succes te meten. Men kan het aantal juist gesplitste woorden nemen,of het aantal juiste splitsingen. Men kan dit doen voor de lopende woorden of voor de verschillende woorden. In ons proefmateriaal was het aantal fout gesplitste woorden gelijk aan het aantal foute splitsingen. Wij vonden de volgende aantallen fouten:

64 van de 4114 verschillende woorden fout gesplitst: 1.6 procent.
224 van de 43712 lopende woorden fout gesplitst: 0.5 procent
224 van de 28229 koppeltekens fout aangebracht: 0.8 procent.

De 64 gevonden fouten zijn op drie manieren te verdelen:

1. Naar het aantal medeklinkers tussen twee klinkers

VCV	22	DOE-LEINDEN
VCCV	16	KO-PLAMP
V..CCCV	26	BEROEP-STROTS

2. Samenstellingen.

Samenstellingen waarvan tweede lid begint met een klinker	22	SLACH-TOFFER
Samenstellingen met S bij de splitsing betrokken	15	KOER-SPEIL
Andere samenstellingen	12	WEREL-DRECORD
Totaal samenstellingen:	49	
Fouten door voorvoegsels	11	AF-RIKA
Fouten door achtervoegsels	2	VERKL-AARD
Fouten door EE	2	HEER-ENVEEN
Totaal geen samenstellingen	15	
3. Een S bij de splitsing betrokken	21	
geen S bij de splitsing betrokken	43	

Hieruit blijkt dus dat het gevaarlijkste de samengestelde woorden zijn (dat een woord samengesteld is is alleen vast te stellen op grond van grotere lengte) en de woorden met een S. Wanneer men dus liever geen splitsing heeft dan een foutieve,zou men ,als er een S in de buurt is,niet moeten splitsen.

Een voorbeeld van een door SYLSPLITS behandelde tekst:

Kwarts-la-gen ma-ken kwarts-la-gen. De twee-de zee-man veeg-de met een mee-ge-no-men zeem-lap leem-aar-de van het tweed-kleed-op-per-vlak. Pre-mier De Quay geeft freu-le Wtte-waal eau-de-co-log-ne en een skij-um-per. de he-ren Van Eijs-den, Krae-mer, Baay-en en De Bruyn her-in-ne-ren zich de hors d'oeu-vre. De bes-te ges-te bij ge-ste-gen be-ste-din-gen is een be-zui-ni-gings-actie. De woes-te toe-stand van de vloei-stof on-der de mi-cro-scoop was een on-ont-koom-baar suc-ces op de in-ter-na-ti-o-na-le pers-con-fe-ren-tie.

3.4 Toepassingen

3.4.1 De belangrijkste toepassing van de lettergreepscheiding is het automatisch uitlijnen van teksten. Het zetten van op een flexowriter getypte kopij kan geheel automatisch gebeuren als het afbreken aan het eind van de regel geautomatiseerd is. Speciaal voor kranten met hun smalle kolombreedte en hoge snelheidseisen biedt dit practische perspectieven. Dit rapport werd door de Electrologica X1 automatisch persklaar gemaakt,rekening houdend met een gegeven bladspiegel en de variabele letterbreedte van de flexowriter "President". Als illustratie werd deze pagina in twee kolommen afgedrukt. Het probleem van de lettergreepscheiding is in deze toepassing iets anders te stellen: men hoeft niet meer te weten waar alle koppeltekens in een woord geplaatst moeten worden,maar het is voldoende om het meest verantwoorde koppelteken te kunnen plaatsen. Zo kan men b.v. de S trachten te vermijden. Ook kunnen voorvoegsels en achtervoegsels worden toegevoegd die niet afgesplitst worden,maar alleen prohibitief werken met de waarschuwing: een splitsing in deze buurt is riskant.

De bovengenoemde foutenpercentages kunnen dus door deze strategie verlaagd worden , maar anderzijds komen lange woorden eerder in aanmerking om gesplitst te worden,en dat zijn juist vaak de samenstellingen. Experimenteel bleek dat in een boekje [9] met 320 regels met afgebroken woorden er drie fouten optraden. In dit rapport zijn 65 regels afgebroken, waarvan 0 foutief.

3.4.2. Een inventarisatie van de Nederlandse lettergrepen met hun frekwenties wordt nu mogelijk,en een onderzoek naar de manier waarop de Nederlandse woorden uit die lettergrepen zijn opgebouwd.

3.5. Opmerkingen

Bij het programma werd alleen met het Nederlands rekening gehouden. Een aanpassing voor b.v. het Duits lijkt mogelijk. SYLSPLITS gaf met teksten uit andere talen het volgende resultaat:

The Ma-the-ma-ti-cal Cen-tre at Am-ster-dam, foun-ded the 11th of Febru-a-ry 1946, is a non-pro-fit in-sti-tu-ti-on ai-ming at the pro-mo-ti-on of pu-re ma-the-ma-tics and its ap-pli-ca-ti-ons.

La ma-ni-pu-la-ti-on au-to-ma-ti-que des in-for-ma-ti-ons con-sti-tuant un cha-pi-tre im-por-tant de l'Au-to-ma-ti-que, il nous a pa-ru u-ti-le d'e-vo-quer les do-mai-nes prin-ci-paux de l'Au-to-ma-ti-que et de rappe-ler en ou-tre la sig-ni-fi-ca-ti-on du mot in-for-ma-ti-on.

Die auf-ga-be die-ser Ar-beit ist es, das ma-the-ma-ti-sche Ge-setz her-zu-lei-ten, das die Bil-dung der Wor-ter aus Sil-ben be-herrscht.

Le ap-pli-ca-zi-o-ni con-ti-nu-a-men-te cre-scen-ti del-la Ma-te-ma-ti-ca a va-ri ra-mi del-la Scien-za e del-la Tec-ni-ca e-si-go-no l'u-so di mez-zi sem-pre pi-u al-ti di Ma-te-ma-ti-ca an-che da par-te di chi non si de-di-ca ad es-sa ex-pro-fes-so.

Med det-ta be-trak-tel-se-satt blir satt-mas-kin-rem-san en bi-pro-dukt som er-sat-ter an-dra uts-krif-ter vid da-ta-mas-ki-nen.

4. Conclusie

Samengestelde woorden vormen de grootste foutenbron bij het automatisch scheiden van lettergrepen,en ze zullen dat blijven doen bij iedere verfijning van het programma. Daar het proces van Reifler [10] om samengestelde woorden te ontleden met behulp van een lijst van alle enkelvoudige woorden voor het Nederlands waarschijnlijk ook effectief is,is een volledige oplossing van het lettergreepprobleem pas mogelijk door als informatie alle enkelvoudige woorden mee te geven. Dus een groot gedeelte van het hele woordenboek,maar nu zonder koppeltekens tussen de lettergrepen zoals bij de in de inleiding genoemde triviale oplossing.

Het automatisch tellen van de lettergrepen in Nederlandse woorden mag als opgelost worden beschouwd. Het succespercentage van 99 procent bij het automatisch verdelen in lettergrepen is hoog genoeg om practische toepassingen daarvan mogelijk te maken.

5. Literatuur

[1] Woordenlijst van de Nederlandse taal,'s-Gravenhage 1954. Leidraad IV behandelt de verdeling in lettergrepen.

[2] Duncan e.a.,Computer Typesetting:an evaluation of the problems Ferranti Computer World,July-August 1963

[3] G.Bafour,A new method for Text Composition: the BBR system Printing Tech.Vol V-10

[4] H.Werner,Elektronisk datamaskin,snabb tabelrattare, Grafisk Forum nr. 11,1962

[5] Weekblad TIME van 18 januari 1963.

[6] P.Naur e.a.,Revised Report on the Algorithmical Language ALGOL 60, Copenhagen 1962. (Numerische Mathematik 4 (1963),420,Communications of the ACM 6 (1963),1,The Computer Journal 5 (1963),349)

[7] J.A.Th.M.v.Berckel,Onderzoek woordfrequentie, Resultaten kranten,Rapport Rekenafdeling Mathematisch Centrum R642/2.

[8] W.Fucks, Theorie der Wortbildung,Math.Phys.Semesterberichte IV 1954.

[9] Uitgave ter gelegenheid van het eerste lustrum van het Nederlands Rekenmachine Genootschap.

[10] E.Reifler,Mechanical determination of the constituents of German substantive compounds, Mech.Transl. 2,1.

SUMMARY

The counting of syllables and the splitting into syllables of Dutch words by a computer are discussed. Both problems are solved by a program written in ALGOL 60. The automatic splitting into syllables is succesful for 99.5 percent of the running words in Dutch newspaper prose. The main application of this research lies in the possibility of automatic typesetting. The lines of this report were justified by the Electrologica X1 of the Mathematical Centre in Amsterdam.