

**stichting  
mathematisch  
centrum**



---

REKENAFDELING

NR 28/72

JUNI

J.C. VAN VLIET  
BESTESPLITS 1  
EEN PROCEDURE VOOR HET AUTOMATISCH  
AFBREKEN VAN NEDERLANDSE WOORDEN

Vertrouwelijk

---

**2e boerhaavestraat 49 amsterdam**

Inhoud:

0. Inleiding.
  1. Voorbereidende werkzaamheden.
  2. Eerste splitsingsfase.
  3. Tweede splitsingsfase.
  4. Derde splitsingsfase.
  5. Resultaten.
  6. Opmerkingen.
  7. Tekst van de procedure.
  8. Vaste getalband.
  9. Voorbeeld.
- Referenties.

## 0. Inleiding.

In dit rapport wordt een algoritme beschreven die de meest betrouwbare plaats bepaalt om een woord af te breken. Hierbij wordt alleen naar het woord zelf gekeken en niet naar de context. De officiële regels voor het afbreken van woorden, zoals gegeven in [1], zijn slechts zeer ten dele bruikbaar voor automatische verwerking.

Als we een "klinkergroep" definiëren als een groep opeenvolgende klinkers die als eenklank of tweeklank uitgesproken kan worden, dan kunnen we in het algemeen stellen dat een woord ergens tussen twee opeenvolgende klinkergroepen afgebroken kan worden. Het aantal lettergrepen van een woord is dus in principe gelijk aan het aantal klinkergroepen. Wanneer we nu een klinkergroep door een V en een medeklinker (inbegrepen de combinaties ch en qu) door een C voorstellen, kunnen we drie gevallen onderscheiden:

1. VV: Hiertussen wordt niet gesplitst, gedeeltelijk uit esthetische overwegingen, gedeeltelijk vanwege het vrij grote aantal fouten dat hierbij op kan treden, zoals bij zoiets (zoi-ets). Een groot aantal hierbij gemaakte fouten kan mogelijk verklaard worden door het tegenwoordig vrij frequent voorkomen van vreemde woorden. (team, research, coach, dashboard).
2. VCV: Behoudens een aantal uitzonderingen (noga!) kan hier gesplitst worden volgens V-CV, in enkele andere gevallen volgens VC-V.
3. Meer dan één medeklinker tussen twee opeenvolgende klinkergroepen. Hier doen zich verschillende mogelijkheden voor:
 

|                                   |               |
|-----------------------------------|---------------|
| splitsen tussen twee medeklinkers | (ijs-breker)  |
| splitsen vóór twee medeklinkers   | (se-cretaris) |
| niet splitsen                     | (kwartslagen) |

Als algemene regel is verder aangehouden dat er niet direct na de eerste of voor de laatste letter gesplitst wordt.

Bij de ontwikkeling van de algoritme is dankbaar gebruik gemaakt van de resultaten uit [2] en van de procedure *bestesplits* van Brandt Corstius[3]. De algoritme is, evenals die van Brandt Corstius, grotendeels gebaseerd op het krantenmateriaal uit [4].

Verschillen ten opzichte van de procedure van Brandt Corstius zijn:

1. Er wordt niet tussen twee klinkergroepen gesplitst als er geen medeklinker(s) tussen staat (staan).
2. Er wordt geen gebruik gemaakt van achtervoegsels, aangezien dit geen verbetering met zich blijkt te brengen.
3. Slechts een gedeelte van de voorvoegsels wordt gebruikt, zodat het aantal juiste splitsingen verhoogd wordt.
4. Het zoeken van de splitsing geschiedt in drie fasen, waarbij in de eerste fase splitsingen gezocht worden met een zeer geringe kans op fouten, terwijl pas in de tweede en derde fase splitsingen aan bod komen met een iets grotere kans op fouten. Hierbij zijn de splitsingen uit de tweede fase in het algemeen esthetisch iets bevredigender dan die uit de derde fase (zoals splitsingen tussen twee delen van een samenstelling). Bij elk van deze drie fasen wordt het woord van achteren af onderzocht, zodat een splitsing gevonden wordt die zover mogelijk naar achteren ligt. Dit geeft een zo goed mogelijke uitvulgraad wanneer de algoritme toegepast wordt om tekst uit te lijnen.
5. Aan de hand van nieuw krantenmateriaal uit 1971 zijn nog enkele verbeteringen aangebracht.

De algoritme is in twee versies beschikbaar: als procedure in ALGOL 60 [5] (in dit rapport afgedrukt) en als function in FORTRAN IV [6].

De auteur is dank verschuldigd aan L.G.L.T. Meertens voor de vele waardevolle suggesties die hij van hem mocht ontvangen.

1. Voorbereidende werkzaamheden.

De procedure *bestesplits1* maakt gebruik van vier integer arrays, die in een omliggend programma gedeclareerd en gevuld dienen te worden:

1. *r*, een array waarin het te splitsen woord dient te staan tussen twee indices, zeg *eerste* en *laatste*. Een mogelijke aanroep van *bestesplits1* luidt nu:

*1 plaats := bestesplits1 (eerste, laatste)*

Na afloop geeft *plaats* de index van het symbool in *r* aan waarachter gesplitst kan worden. Als *plaats* gelijk is aan *eerste* - 1 is er geen splitsing gevonden.

2. *vertaal*, met behulp waarvan de symbolen uit *r* gehercodeerd worden.
3. *comprimatie*, waarmee combinaties van opeenvolgende letters zo mogelijk gecomprimeerd worden.
4. *consonant*, een array waarin alle benodigde informatie staat voor splitsing bij twee of meer opeenvolgende medeklinkers.

Een voorbeeld van het gebruik van de procedure *bestesplits1* is nu (zie voor de standaard-procedure *read* bijvoorbeeld [7]):

```
begin integer array r [1:50], vertaal [0:127], comprimatie [26:40, 29:42],  
                    consonant [39:61, 39:61];
```

```
integer i, j, bovengrens, plaats;
```

```
integer procedure bestesplits1 (eerste, laatste); value eerste,  
                    laatste; integer eerste, laatste;
```

```
comment voor de body van deze procedure zie 7;
```

```
boolean procedure lees woord (bovengrens); integer bovengrens;
```

```
comment deze procedure leest een woord en plaatst het in het  
                    array "r", "bovengrens" krijgt als waarde de index van het  
                    laatste symbool van het woord in "r".
```

```
                    Als er geen invoer meer is, wordt lees woord false gemaakt;
```

```
procedure druk woord af (plaats); value plaats; integer plaats;
```

```
comment deze procedure drukt het woord af, met na het symbool  
                    dat op index "plaats" staat, een koppelteken;
```

```

for i:= 0 step 1 until 127 do vertaal [i]:= read;
for i:= 26 step 1 until 40 do
for j:= 29 step 1 until 42 do comprimatie [i,j]:= read;
for i:= 39 step 1 until 61 do
for j:= 39 step 1 until 61 do consonant [i,j]:= read;

for i:= 0 while lees woord (bovengrens) do
begin plaats:= bestesplits1 (1, bovengrens);
           if plaats > 0 then druk woord af (plaats)
end
end

```

Bij een aanroep van de procedure *bestesplits1* gebeurt, alvorens de eigenlijke analyse van het woord plaatsvindt, het volgende:

a) vertaling van het woord in een interne codering met behulp van het array *vertaal*. De vulling van het array *vertaal* moet zodanig zijn, dat alle enkele letters (inclusief het trema) een waarde krijgen overeenkomstig de specificatie uit onderstaande tabel, en alle andere symbolen een waarde groter dan 64. In de versie die door ons is gebruikt hebben alle symbolen een waarde tussen 0 en 127 (vandaar de declaratie *vertaal* [0:127]). Wanneer bijvoorbeeld de letter a in het array *r* de waarde 10 heeft, moet *vertaal* [10] gelijk zijn aan 29. (Er wordt binnen de procedure geen onderscheid gemaakt tussen hoofdletters en kleine letters, dus wanneer de letter A de waarde 37 heeft, moet eveneens gelden dat *vertaal* [37] gelijk is aan 29.)

Wanneer er een andere codering voor de invoer gebruikt wordt, zal dus de vulling van het array *vertaal* moeten worden aangepast. De vulling van de arrays *comprimatie* en *consonant* hoeft dan niet veranderd te worden. (De grenzen van de arrays *comprimatie* en *consonant* moeten steeds gelijk zijn aan die uit bovenstaand voorbeeld. De grenzen van het array *vertaal* kunnen per systeem variëren. Voor de vulling van de arrays: zie 8.

b) comprimeren (met behulp van het array *comprimatie*).

c) bepalen van het aantal lettergrepen en de plaatsen waar klinkergroepen voorkomen.

Na afloop hiervan staat het woord in het array *woord*, waarbij de diverse symbolen de volgende waarden hebben gekregen:

|    |     |    |     |    |    |    |            |
|----|-----|----|-----|----|----|----|------------|
| 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  | aa  | 23 | --  | 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 | .. (trema) |
| 12 | eeu | 28 | oo  | 44 | b  | 60 | qu         |
| 13 | ei  | 29 | a   | 45 | d  | 61 | ch         |
| 14 | eui | 30 | e   | 46 | f  | 62 | --         |
| 15 | ey  | 31 | i   | 47 | g  | 63 | --         |
| 16 | ieu | 32 | u   | 48 | l  | 64 | --         |

Deze codering is niet essentieel, maar maakt het programma iets overzichtelijker en sneller. In deze codering (die in de arrays *vertaal* en *comprimatie* staat) kan het trema aangegeven worden door een onderstreping. Wanneer men niet bij een trema wil splitsen kan men in het array *vertaal* de codering hiervoor gelijk maken aan die van de letter x (of een waarde groter dan 64). Er moet echter wel rekening mee worden gehouden dat bij eventuele splitsing voor een trema, dit trema door de gebruiker verwijderd dient te worden! Meer bijzonderheden omtrent deze voorbereidingsfase zijn te vinden in [2].

2. Eerste splitsingsfase.

In de eerste splitsingsfase wordt gekeken of er, tussen twee klinkergroepen, opeenvolgende medeklinkers staan waartussen zeker gesplitst kan worden. Het blijkt dat dit bij de overgrote meerderheid van de combinaties het geval is. Inplaats van de lijst van combinaties te geven, waartussen wel gesplitst kan worden, wordt daarom de lijst van combinaties gegeven waartussen in de eerste fase juist niet gesplitst wordt. Deze luidt:

|     |     |     |    |    |    |     |     |
|-----|-----|-----|----|----|----|-----|-----|
| ck  | cl  | cr  |    |    |    |     |     |
| hm  | hn  |     |    |    |    |     |     |
| kl  | kn  | kr  | ks | kw |    |     |     |
| bl  | br  |     |    |    |    |     |     |
| dr  | ds  | dw  |    |    |    |     |     |
| fd  | fl  | fr  | fs | ft |    |     |     |
| gh  | gd  | gl  | gr | gs |    |     |     |
| lk  | ld  | lf  | lg | lm | lp | ls  |     |
| md  | ms  |     |    |    |    |     |     |
| nk  | nd  | ng  | ns | nt |    |     |     |
| pj  | ph  | pl  | pr | ps |    |     |     |
| rk  | rd  | rg  | rm | rn | rp | rs  |     |
| sh  | sl  | sm  | sn | sp | st | squ | sch |
| tj  | th  | tr  | ts | tw |    |     |     |
| vl  | vr  |     |    |    |    |     |     |
| wr  |     |     |    |    |    |     |     |
| zw  |     |     |    |    |    |     |     |
| chl | chr | cht |    |    |    |     |     |

Bij al de combinaties (ook die uit de tweede splitsingsfase) waarbij gesplitst kan worden geldt echter dat niet tot splitsing wordt overgegaan wanneer er zo een lettergreep zou ontstaan met aan het begin een niet uitspreekbare combinatie van medeklinkers (brunntal wordt in deze fase dus niet gesplitst hoewel normaal in deze fase tussen n en n gesplitst wordt).



### 3. Tweede splitsingsfase.

In de tweede splitsingsfase wordt nogmaals het woord doorlopen en wordt mogelijk alsnog bij twee opeenvolgende medeklinkers gesplitst. De zaak ligt hier echter iets gecompliceerder. We kunnen drie gevallen onderscheiden:

- a) Splitsen tussen twee medeklinkers. Hieronder vallen enkele combinaties die iets minder "zeker" zijn dan die uit de eerste splitsingsfase:

hm hn

gh

ms

pj

Terwijl de combinaties uit de eerste splitsingsfase grotendeels bepaald zijn aan de hand van de woordgrammatica uit [2], ligt de zaak hier moeilijker. De combinaties zijn empirisch tot stand gekomen. Opgenomen zijn die, welke zo weinig mogelijk fouten tot gevolg hebben.

- b) Splitsen bij een "tweecons" (twee opeenvolgende medeklinkers die samen uitgesproken kunnen worden). Hierbij onderscheiden we weer twee gevallen:

i. VCCLV, L staat voor 0, 1 of meer medeklinkers, waarbij niet tot splitsing is overgegaan. Hier wordt gesplitst volgens V-CCLV (bijvoorbeeld ge-zworen).

ii.  $C_1C_2C_3LV$ , L staat voor 0, 1 of meer medeklinkers, waarbij niet tot splitsing is overgegaan en  $C_2C_3$  is de onderzochte tweecons. Hier wordt gesplitst volgens  $C_1-C_2C_3LV$  als de combinatie  $C_1C_2$  onder a) valt, en  $-C_1C_2C_3LV$  als de combinatie  $C_1C_2$  onder b) valt. (voorbeelden hiervan zijn gram-schap respectievelijk ge-schrift.)

De combinaties die onder tweecons gerekend worden, zijn:

cl cr

kl kr kw

bl br

dr  
 pl pr  
 sch  
 vl vr  
 wr  
 zw  
 chl chr

- c) Hieronder vallen een aantal combinaties waarbij alleen gesplitst wordt in het geval V...CCV, en wel volgens V...C-CV (zo wordt koningin gesplitst als konin-gin, doch koningshuis wordt niet gesplitst). De combinaties waarop dit van toepassing is, zijn:

ck  
 fd ft  
 gd  
 lk ld lf lg lm lp ls  
 md  
 nk nd ng ns nt  
 rk rd rg rm rn rp  
 ts  
 cht

In alle overige gevallen wordt niet gesplitst, vanwege het relatief grote aantal fouten dat hierbij kan optreden. Het betreft de combinaties:

kn ks  
 ds dw  
 fl fr fs  
 gl gr gs  
 ph ps  
 rs  
 sh sl sm sn sp st squ  
 tj th tr tw

#### 4. Derde splitsingsfase.

In deze derde en laatste fase wordt onderzocht of er alsnog een mogelijkheid is om het woord te splitsen, en wel in het geval  $V_1CV_2$ . Dit gaat in de volgende stappen:

- i. Is C een van de medeklinkers j, v of z, splits dan volgens  $V_1-CV_2$  (bijvoorbeeld loo-zen).
- ii. Is C de medeklinker x, splits dan niet.
- iii. Is  $V_1$  de klinkergroep ee splits dan ook niet (hierbij kunnen veel fouten optreden, b.v. meetellen versus meetapparaat).
- iv. Is  $V_1$  een van de klinkergroepen aa, oo of uu, splits dan volgens  $V_1C-V_2$ .
- v. Zij L de letter voor  $V_1$ , splits dan niet in de gevallen dat  $LV_1C$  gelijk is aan: ven, ter, nog, der, ver, hier of van, en splits anders volgens  $V_1-CV_2$  (bij de genoemde voorvoegsels ontstaan onoverkomelijke moeilijkheden, zoals bij veren versus verenigd).

5. Resultaten.

Deze procedure is ontwikkeld en getest aan de hand van een gedeelte van het bestand uit [3] (alle woorden die meer dan eenmaal voorkomen). Hierna is de procedure gebruikt voor een hoeveelheid krantenmateriaal uit juni 1971 en aan de hand hiervan op sommige plaatsen herzien. In de volgende tabel zijn de bereikte resultaten weergegeven, waarbij alleen de meer-lettergreppige woorden opgenomen zijn:

|                | oud bestand    | nieuw bestand  | totaal         |
|----------------|----------------|----------------|----------------|
| goed gesplitst | 13181 (90.64%) | 15052 (90.43%) | 28233 (90.52%) |
| fout gesplitst | 75 ( 0.51%)    | 27 ( 0.16%)    | 102 ( 0.33%)   |
| niet gesplitst | 1287 ( 8.85%)  | 1567 ( 9.41%)  | 2854 ( 9.15%)  |
| totaal         | 14543 (100%)   | 16646 ( 100%)  | 31189 (100%)   |

De lijst van gemaakte fouten ziet er als volgt uit (eerst worden die uit het oude bestand [3] vermeld, vervolgens die uit het nieuwe bestand (1971), gesplitst naar de drie splitsingsfasen, het getal achter het woord geeft steeds de frequentie aan).

oud bestand

| eerste fase   |   | tweede fase    |   | derde fase |    |
|---------------|---|----------------|---|------------|----|
| chroests-jev  | 2 | al-sof         | 2 | ade-nauers | 2  |
| chroests-jews | 2 | fil-minstituut | 2 | zenu-warts | 2  |
| khroests-jev  | 2 |                |   | sha-pe     | 4  |
| chroests-jow  | 3 |                |   | ade-nauer  | 25 |
| kroes-jefs    | 3 |                |   |            |    |
| mar-xisme     | 3 |                |   |            |    |
| chroests-jew  | 4 |                |   |            |    |
| chroesjts-jew | 5 |                |   |            |    |
| chroesjts-jow | 7 |                |   |            |    |
| kroes-jef     | 7 |                |   |            |    |

nieuw bestand

| eerste fase   |   | tweede fase       |   | derde fase  |   |
|---------------|---|-------------------|---|-------------|---|
| hats-jiekadee | 1 | al-sook           | 1 | diere-narts | 1 |
| slots-cene    | 1 | kritie-kloos      | 1 | sto-koud    | 1 |
| kor-tom       | 2 | macht-soverdracht | 1 | vo-lop      | 1 |
|               |   | mic-ky            | 1 | we-leens    | 1 |
|               |   | nic-ky            | 1 | feye-noord  | 2 |
|               |   | rech-top          | 1 |             |   |
|               |   | ron-dom           | 1 |             |   |
|               |   | ron-duit          | 1 |             |   |
|               |   | staat-samateurs   | 1 |             |   |
|               |   | stoe-prand        | 1 |             |   |
|               |   | di-kwijls         | 2 |             |   |
|               |   | jac-ko            | 2 |             |   |
|               |   | al-sof            | 3 |             |   |

Een analyse van deze fouten levert het volgende op:

- a) in de eerste splitsingsfase werden 42 fouten gemaakt (41%). De meeste hiervan traden op bij vreemde woorden (de zeven manieren waarop XPYLLQEB geschreven kan worden).
- b) In de tweede splitsingsfase werden 21 fouten gemaakt (21%). Hiervan waren er 4 fout omdat ten onrechte een tweecons gevonden werd. De rest valt onder hetgeen onder 3.c besproken is. Deze kunnen voorkomen worden door in dat geval alleen te splitsen als de volgende klinkergroep gelijk is aan e of i. Hierdoor neemt echter ook het aantal splitsingen af met circa 3%. In dit geval moeten regel 102 en 103 van de procedure (zie 7) veranderd worden in:

else if abs (hulp) = 4 ^

(woord [klinker 2] = 30 v woord [klinker 2] = 31)

then splits af na (i);

- c) Van de 102 gemaakte fouten zijn er 70 gemaakt in niet-nederlandse woorden ("echt" fout gaat dus slechts.10%).

De met deze procedure bereikte resultaten tonen aan dat het mogelijk is deze algoritme te gebruiken voor het mechanisch uitlijnen van teksten. Zo valt te schatten dat gemiddeld eens op de twee krantepagina's een afbreekfout gemaakt zal worden. (Waarschijnlijk zelfs minder omdat in het algemeen lange woorden, die bij deze procedure veiliger gesplitst worden, een grotere kans maken afgebroken te moeten worden dan kortere woorden. Het werkelijke foutenpercentage zal dan nog lager komen te liggen.)

Ter vergelijking kan nog het volgende dienen: de procedure *bestesplits* van Brandt Corstius splitste in totaal 21812 woorden (69.9%) en maakte daarbij 250 fouten (1.15%). Van deze fouten traden er 68 op in vreemde woorden.

6. Opmerkingen.

1. Eerder in dit rapport is reeds opgemerkt dat een splitsing alleen geaccepteerd wordt wanneer de dan volgende lettergreep begint met een daar uitspreekbare medeklinkercombinatie ("u.m.c."). De hierdoor voorkomen fouten zijn misschien niet groot in aantal, maar behoren wel tot de meest in het oog springende.

Het is interessant na te gaan op welke wijze deze splitsingen voorkomen worden.

Allereerst zullen we het volgende aantonen:

Zij X een aan het begin van een lettergreep onuitspreekbare medeklinkercombinatie ("o.m.c.") en Y een medeklinkercombinatie die X bevat. Dan is ook Y een o.m.c.

Bewijs (door inductie naar het aantal letters n dat in Y aan X voorafgaat):

- i. Zij  $n = 0$ . Een lettergreep die met Y begint, begint dan ook met X en is derhalve onuitspreekbaar.
- ii. Zij  $n > 0$ . Veronderstel dat Y uitspreekbaar is. Op eenvoudige wijze valt aan de hand van de in [2] genoemde lijst van aan het begin van een lettergreep voorkomende medeklinkercombinaties na te gaan dat al deze combinaties uitspreekbaar blijven, wanneer daarvan de eerste letter wordt afgehaald. Dit moet dan ook voor X gelden; dit is dan echter in strijd met de inductieveronderstelling.

Zo gauw nu bij het onderzoek van een medeklinkercombinatie (dat van rechts naar links plaatsvindt) een o.m.c. (van twee letters) wordt aangetroffen (zonder dat hierbij gesplitst wordt), dan wordt het verdere onderzoek van deze medeklinkercombinatie gestaakt. Dit is terecht aangezien toch iedere eventueel gevonden splitsing onderdrukt zou moeten worden. Verder is op eenvoudige wijze het volgende na te gaan:

laat ps, anders dan in [2], als o.m.c. gerekend worden. Dan geldt i.h.a.: als X, Y en Z zodanige medeklinkercombinaties zijn, dat zowel XY als YZ een u.m.c. is, dan is ook XYZ een u.m.c. (een uitzondering is b.v.:  $X = s$ ,  $Y = ch$ ,  $Z = \}$ ).

Wanneer wij deze uitzonderingen vanwege hun zeldzaamheid van voorkomen buiten beschouwing laten, dan volgt nu dat bij het onderzoek naar de uitspreekbaarheid van een medeklinkercombinatie na een overwogen splitsing alleen de laatste twee in behandeling genomen letters in beschouwing genomen behoeven te worden. (In de matrix *consonant* zijn o.m.c.'s en u.m.c.'s aangegeven door een negatief respectievelijk positief getal. De o.m.c.'s zijn bepaald aan de hand van de woordgrammatica uit [2], met toevoeging van enkele veel fouten veroorzakende combinaties, zoals st en ps).

2. De procedure *bestesplits1* kan voor het uitlijnen van teksten op de volgende wijze gebruikt worden: stel een gegeven woord past niet meer op de regel. Bied nu dit woord aan de procedure *bestesplits1* aan. Vindt *bestesplits1* geen splitsing, lijn dan de regel uit en plaats het woord in zijn geheel op de volgende regel. Vindt *bestesplits1* wel een splitsing en past het eerste gedeelte (inclusief het koppelteken) nog op de regel, plaats dan dit deel op de regel, lijn deze regel uit en plaats de rest van het woord op een nieuwe regel. Wanneer het eerste gedeelte niet op de regel past, kan inplaats van het gehele woord alleen het eerste gedeelte plus nog een extra letter van het tweede gedeelte aan *bestesplits1* aangeboden worden. Dit laatste is nodig omdat *bestesplits1* weigert één letter aan het eind af te splitsen, zodat na vuuro-ven de splitsing vuur-oven ten onrechte niet meer gevonden zou worden. Vanwege dezelfde eigenschap van *bestesplits1* is de gegeven oplossing veilig; er worden immers minstens twee letters afgesplitst. Dit proces kan herhaald worden totdat hetzij een splitsing gevonden wordt waarbij het gedeelte voor de splitsing nog op de regel past, hetzij *bestesplits1* geen splitsing meer vindt. (Het effect van het aanbieden van gedeelten van woorden op het foutenpercentage is door ons niet onderzocht.)



7. Tekst van de procedure.

```

1  integer procedure bestesplits1 (eerste, laatste);
2  value eerste, laatste; integer eerste, laatste;
3  begin integer i, j, beurt index, verschil, hulp, nivo,
4      letter, volgende letter, aantal syllaben, aantal,
5      klinker 1, klinker 2,
6      ven, ter, nog, der, ver, hier, van,
7      letter ee, letter j, letter v, letter x, letter z,
8      start 1, start 2, stop;
9      integer array woord, verwijzing[0 : 50], klinker,
10     beurt[1 : 20];
11
12     procedure splits af na(h); value h; integer h;
13     begin if h < 2 then goto
14         (if nivo = 3 then geen splitsing else volgende);
15         bestesplits1:= h:= verwijzing[h + 1] + eerste - 2;
16         goto (if h < laatste - 1 ^ h > eerste then exit else
17             geen splitsing)
18     end splits af na;
19
20 initialisatie:
21     ven:= 227 250; ter:= 223 156; nog:= 206 959; der:= 186 292;
22     ver:= 227 252; hier:= 174 324; van:= 227 186;
23     letter ee:= 27; letter j:= 41; letter v:= 55; letter x:= 57;
24     letter z:= 58;
25 vertalen in interne codering:
26     woord[0]:= 0; i:= j:= 1;
27     for hulp:= eerste step 1 until laatste do
28     begin letter:= r[hulp];
29         letter:= if letter > 0 ^ letter < 128 then
30         vertaal[letter] else 63;
31         if letter < 63 then
32         begin woord[j]:= letter; verwijzing[j]:= i; j:= j + 1
33         end;
34         i:= i + 1
35     end vertalen;
36     aantal:= j - 1; if aantal < 4 then goto geen splitsing;
37     i:= j:= 1; hulp:= 0; goto opgebouwd;

```

```

38 comprimatie van klinkers:
39   if letter > 25 ^ letter < 41 then
40     begin volgende letter := woord[i + 1];
41     if volgende letter > 28 ^ volgende letter < 43 then
42       begin letter := comprimatie[letter, volgende letter];
43       if letter > 0 then
44         begin i := i + 1; hulp := hulp + 1; woord[i] := letter;
45         goto opgebouwd
46       end
47     end
48   end comprimatie;
49   woord[j] := woord[i]; j := j + 1; i := i + 1;
50   verwijzing[j] := verwijzing[j] + hulp;
51 opgebouwd:
52   if i ≠ aantal then
53     begin letter := woord[i]; goto comprimatie van klinkers end;
54 bepaling aantal syllaben:
55   woord[j] := woord[aantal]; aantal := j; j := 1;
56   for i := 1 step 1 until aantal do
57     if woord[i] < 39 then
58       begin klinker[j] := i; j := j + 1 end;
59     aantal syllaben := j - 1;
60     if aantal syllaben < 2 then goto geen splitsing;
61     beurt index := verschil := start 1 := 1;
62     start 2 := aantal syllaben;
63 weer:
64     for i := aantal syllaben step - 1 until 2 do
65       if klinker[i] - klinker[i - 1] = verschil then
66         begin beurt[beurt index] := i - 1;
67         beurt index := beurt index + 1
68       end;
69       if verschil = 1 then start 1 := beurt index else
70       if verschil = 2 then start 2 := beurt index;
71       if beurt index < aantal syllaben then
72         begin verschil := verschil + 1; goto weer end;
73       beurt index := start 2 - 1; stop := aantal syllaben; nivo := 1;

```

```

74 volgende:
75 beurt index:= beurt index + 1; if beurt index > stop then
76 begin if nivo = 1 then beurt index:= start 2 else
77   if nivo = 2 then
78     begin beurt index:= start 1; stop:= start 2 end
79     else goto geen splitsing;
80     nivo:= nivo + 1;
81     if beurt index > stop then goto volgende
82   end;
83   klinker 1:= klinker[beurt[beurt index]];
84   klinker 2:= klinker[beurt[beurt index] + 1];
85   if nivo = 1 then
86     begin for i:= klinker 2 - 2 step - 1 until klinker 1 + 1 do
87       begin hulp:= consonant[woord[i], woord[i + 1]];
88         if abs(hulp) = 1 then splits af na(i);
89         if hulp < 0 then goto volgende
90       end
91     end eerste fase
92   else if nivo = 2 then
93     begin for i:= klinker 2 - 2 step - 1 until klinker 1 + 1 do
94       begin hulp:= consonant[woord[i], woord[i + 1]];
95         if abs(hulp) = 2 then splits af na (i);
96         if abs(hulp) = 3 then
97           begin if i = klinker 1 + 1 then splits af na (i - 1);
98             hulp:= abs(consonant[woord[i - 1], woord[i]]);
99             if hulp = 2 then splits af na (i - 1);
100            if hulp = 3 then splits af na(i - 2)
101          end
102         else if abs(hulp) = 4 ^
103           i = klinker 2 - 2 then splits af na(i);
104         if hulp < 0 then goto volgende
105       end
106     end tweede fase

```

```

107 else if nivo = 3 then
108 begin hulp := woord[klinker 1 + 1];
109 if hulp = letter j v
110 hulp = letter v v
111 hulp = letter s then splits af na (klinker 1);
112 if hulp = letter w then goto volgende;
113 hulp := woord[klinker 1];
114 if hulp = letter ee then goto volgende;
115 if hulp > 24 ^
116 hulp < 29 then splits af na(klinker 1 + 1);
117 hulp := 4096 × woord[klinker 1 - 1] + 64 × hulp +
118 woord[klinker 1 + 1]
119 if hulp = ven v
120 hulp = ter v
121 hulp = nog v
122 hulp = der v
123 hulp = ver v
124 hulp = hier v
125 hulp = van then goto volgende
126 else splits af na(klinker 1)
127 end derde fase;
128 goto volgende;
129 geen splitsing:
130 bestesplits1 := eerste - 1;
131 exit:
132 end bestesplits1

```

8. Waste getallenband.*vertaal*

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 68 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| 65 | 68 | 29 | 44 | 40 | 45 | 30 | 46 |
| 47 | 42 | 31 | 41 | 43 | 48 | 49 | 50 |
| 33 | 51 | 39 | 52 | 53 | 54 | 32 | 55 |
| 56 | 57 | 38 | 58 | 68 | 29 | 44 | 40 |
| 45 | 30 | 46 | 47 | 42 | 31 | 41 | 43 |
| 48 | 49 | 50 | 33 | 51 | 39 | 52 | 53 |
| 54 | 32 | 55 | 56 | 57 | 38 | 58 | 68 |
| 65 | 64 | 65 | 65 | 65 | 65 | 65 | 65 |
| 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| 65 | 65 | 65 | 65 | 65 | 65 | 65 | 65 |
| 63 | 65 | 65 | 65 | 65 | 65 | 57 | 65 |

*comprimatic*

|    |    |    |    |    |    |   |   |
|----|----|----|----|----|----|---|---|
| 0  | 0  | 1  | 7  | 0  | 0  | 0 | 0 |
| 0  | 2  | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 0  | 12 | 0  | 0  | 0 | 0 |
| 0  | 0  | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 5  | 0  | 0  | 0  | 0 | 0 |
| 0  | 6  | 0  | 0  | 0  | 0  |   |   |
| 26 | 8  | 9  | 10 | 0  | 0  | 0 | 0 |
| 0  | 11 | 0  | 0  | 0  | 0  |   |   |
| -1 | 27 | 13 | 34 | 0  | 0  | 0 | 0 |
| 0  | 15 | 0  | 0  | 0  | 0  |   |   |
| 0  | 35 | 0  | 0  | 0  | 0  | 0 | 0 |
| 0  | 0  | 0  | 0  | 17 | 0  |   |   |
| 0  | 0  | 21 | 25 | 0  | 0  | 0 | 0 |
| 0  | 22 | 0  | 0  | 0  | 0  |   |   |
| 0  | 36 | 18 | 37 | 28 | 0  | 0 | 0 |
| 0  | 20 | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 14 | 0  | 0  | 0  | 0 | 0 |
| 0  | 0  | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 0  | 16 | 0  | 0  | 0 | 0 |
| 0  | 0  | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 3  | 24 | 0  | 0  | 0 | 0 |
| 0  | 4  | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 19 | 0  | 0  | 0  | 0 | 0 |
| 0  | 0  | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 |
| 0  | 0  | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 0  | 60 | 0  | 0  | 0 | 0 |
| 0  | 0  | 0  | 0  | 0  | 0  |   |   |
| 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 |
| 0  | 0  | 0  | 0  | 0  | 61 |   |   |

*consonant*

|    | q  | c  | j  | h  | k  | b  | d  | f  | g  | l  | m  | n  | p  | r  | s  | t  | v  | w  | x  | z  | -  | qu | ch |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| q  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| c  | -1 | -1 | -1 | -1 | -4 | -1 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| j  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| h  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| k  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | +3 | -1 | +5 | -1 | +3 | -5 | -1 | -1 | +3 | -1 | -1 | -1 | -1 | -1 |
| b  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| d  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | +3 | -5 | -1 | -1 | +5 | -1 | -1 | -1 | -1 | -1 | -1 |
| f  | -1 | -1 | -1 | -1 | -1 | -1 | -4 | -1 | -1 | +5 | -1 | -1 | -1 | +5 | -5 | -4 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| g  | -1 | -1 | -1 | +2 | -1 | -1 | -4 | -1 | -1 | +5 | -1 | -1 | -1 | +5 | -5 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| l  | -1 | -1 | -1 | -1 | -4 | -1 | -4 | -4 | -4 | -1 | -4 | -1 | -4 | -1 | -4 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| m  | -1 | -1 | -1 | -1 | -1 | -1 | -4 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -2 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| n  | -1 | -1 | -1 | -1 | -4 | -1 | -4 | -1 | -4 | -1 | -1 | -1 | -1 | -1 | -4 | -4 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| p  | -1 | -1 | +2 | +5 | -1 | -1 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | +3 | -5 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| r  | -1 | -1 | -1 | -1 | -4 | -1 | -4 | -1 | -4 | -1 | -4 | -4 | -4 | -1 | -5 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| s  | -1 | -1 | -1 | -5 | -1 | -1 | -1 | -1 | -1 | +5 | +5 | +5 | +5 | -1 | -1 | +5 | -1 | -1 | -1 | -1 | -1 | +5 | +3 |
| t  | -1 | -1 | -5 | -5 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | +5 | -4 | -1 | -1 | -5 | -1 | -1 | -1 | -1 | -1 |
| v  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| w  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| x  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| z  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | -1 | -1 |
| -  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| qu | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| ch | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | +3 | -1 | -1 | -1 | +3 | -1 | -4 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |

9. Voorbeeld.

| 1 <sup>e</sup> splitsingsfase        | 2 <sup>e</sup> splitsingsfase | 3 <sup>e</sup> splitsingsfase                | niet gesplitst              |
|--------------------------------------|-------------------------------|--|-----------------------------|
| DAVIG-NON<br>DECEM-BER<br>DER-TIG    |                               |  | DESMOND                     |
| DISCUS-SIE                           |                               | DIVI-DEND                                    | DOODSTRAF                   |
| EER-LIJK<br>EGYP-TISCHE<br>EM-MERTJE |                               | DUIDELIJ-KER                                 |                             |
|                                      | EN-KEL                        |  | ERNSTIG<br>ETEN<br>EVENEENS |
|                                      |                               | FAMI-LIE<br>FI-LE                            |                             |
|                                      | FOR-MATIE                     |  |                             |
|                                      |                               | GA-ME<br>GA-TEN<br>GEBIE-DEN<br>GE-BIT       |                             |
|                                      | GE-BLEKEN<br>GE-BLEVEN        |  |                             |
|                                      |                               | GE-DACHT<br>GEGE-VENS<br>GEHE-LE             |                             |
|                                      | GEHOL-PEN                     |  |                             |
|                                      | GEL-DEN                       | GE-HOORD                                     |                             |
| GELEGEN-HEID<br>GELUK-KIG            |                               | GE-LEGD                                      |                             |
|                                      |                               | GENOE-GEN<br>GENO-MEN<br>GE-RICHT<br>GE-RING |                             |
|                                      | GE-SCHREVEN                   |  | GESLAAGD                    |
| GESPREKSPART-NERS                    |                               | GEWEI-GERD                                   |                             |
|                                      | GEWEL-DIGE                    | GEWO-NE                                      |                             |
| GISTERMID-DAG                        |                               | HA-LEN                                       | HAAGSE                      |



Referenties:

- [1] Woordenlijst van de Nederlandse Taal, Staatsdrukkerij- en uitgevers-  
bedrijf/Martinus Nijhoff, 's-Gravenhage, 1954.
- [2] H. Brandt Corstius, Exercises in Computational Linguistics,  
MC Tract 30, Mathematisch Centrum, Amsterdam, 1970.
- [3] H. Brandt Corstius, procedure bestesplits, ongepubliceerd.
- [4] J.A.Th.M. van Berckel e.a., Formal Properties of Newspaper Dutch,  
MC Tract 12, Mathematisch Centrum, Amsterdam, 1965.
- [5] P. Naur (ed.), Revised Report on the Algorithmic Language ALGOL 60,  
Regnecentralen Copenhagen, 1964.
- [6] Control Data 6400/6500/6600 Computer Systems, FORTRAN Extended  
Reference Manual, Sunnyvale, California 94086, publication  
nr 660176600, 1971.
- [7] D. Grune, Handleiding Milli-systeem voor de EL X8, LR 1.1,  
Mathematisch Centrum, Amsterdam, 1970.

Addendum en corrigendum bij rapport NR 28\*)

Bestesplits1

Een procedure voor het automatisch afbreken van  
Nederlandse woorden

door

J.C. van Vliet

\*) Hiermee komt het addendum en corrigendum gedateerd februari 1974  
te vervallen.

## Errata

Op pag. 3 leze men *plaats:= i.p.v. 1 plaats:=*

Op pag. 10 moet staan (90.64%) i.p.v. (90.64%

Op pag. 15 moet de tekst op regel 13-18 luiden:

```
begin bestesplits1:= h:= verwijzing[h + 1] + eerste - 2;  
  goto (if h < laatste - 1 ^ h > eerste then exit else  
  volgende)  
end splits af na;
```

Op dezelfde pagina, regel 36:

I.p.v. *geen spltising* leze men *geen splitsing*

Op pagina 21 staan in kolom 21 en rij 21 (beide gemerkt met een -) getallen -1. Al deze getallen moeten vervangen worden door +5.

Om te voorkomen dat woorden met meer dan 50 letters of meer dan 20 lettergrepen niet verwerkt worden, worden de volgende wijzigingen in het ALGOL-programma voorgesteld:

pag. 15, regel 27:

```
hulp:= eerste -1;  
for hulp:= hulp + 1 while hulp ≤ laatste ^ j ≤ 50 do
```

pag. 16, regel 56:

```
i:= 0;  
for i:= i + 1 while i ≤ aantal ^ j ≤ 20 do
```

Na deze wijzigingen worden van te lange woorden alleen de eerste 50 letters of de eerste 20 lettergrepen in behandeling genomen.

## Aanwijzingen voor het gebruik van de FORTRAN versie van bestesplits1

De FORTRAN versie van bestesplits1 is qua werking en manier van aanroepen identiek aan de ALGOL 60 versie (inclusief de op de vorige pagina voorgestelde wijzigingen). Voordat de function BESTESP aangeroepen kan worden, moet de initialisatie subroutine INIT BSP, die de diverse vaste tabellen inleest, aangeroepen worden. De tabellen VERTAAL, COMPR en CONS, alsmede het array waarin het te splitsen woord staat, moeten (in deze volgorde) door het hoofdprogramma via een COMMON statement aan beide subroutines doorgegeven worden. De routine is, op triviale onderdelen na, geschreven in standaard ANSI FORTRAN. (De I/O statements in INIT BSP zijn niet standaard, en er komen identifiers van meer dan 6 symbolen voor).

```
SUBROUTINE INIT BSP
COMMON VERTAAL(64), COMPR(15, 14), CONS(23, 23)
INTEGER VERTAAL, COMPR, CONS
READ 10, VERTAAL
READ 11, ((COMPR(I,J),J=1,14),I=1,15)
READ 11, ((CONS(I,J),J=1,23),I=1,23)
10 FORMAT(8I10)
11 FORMAT(8(19,1X))
RETURN
END
```

INTEGER FUNCTION BESTESP(EERSTE, LAATSTE)  
COMMON VERTAAL(64), COMPR(15, 14), CONS(23, 23), R(100)  
INTEGER EERSTE, LAATSTE, R, VERTAAL, COMPR, CONS,  
1 I, J, BEURT 1, VERSCH, HULP, NIVO,  
2 LETTER, VOLG LET, AANT SYL, AANTAL,  
3 KLINK 1, KLINK 2,  
4 VEN, TER, NOG, DER, VER, HIER, VAN,  
5 LETT EE, LETT J, LETT V, LETT X, LETT Z,  
6 START 1, START 2, STOP,  
7 WOORD(50), VERW(50), KLINKER(20), BEURT(20),  
8 SPLITS

C  
C INITIALISATIE

C  
DATA VEN, TER, NOG, DER, VER, HIER, VAN  
× /227250, 223156, 206959, 186292, 227252, 174324, 227186/  
× LETT EE, LETT J, LETT V, LETT X, LETT Z/27, 41, 55, 57, 58/

C  
C VERTALEN IN INTERNE CODERING

C  
WOORD(1) = 0  
I = 1  
J = 2  
HULP = EERSTE  
15 LETTER = R(HULP)  
IF(LETTER .GT. 0 .AND. LETTER .LT. 65) GO TO 12  
LETTER = 65  
GO TO 14  
12 LETTER = VERTAAL(LETTER)  
IF(LETTER .GE. 65) GO TO 14  
WOORD(J) = LETTER  
VERW(J) = I  
J = J + 1  
14 I = I + 1  
HULP = HULP + 1  
IF(HULP .LE. LAATSTE .AND. J .LE. 50) GO TO 15  
AANTAL = J - 1  
IF(AANTAL .LT. 5) GO TO 9999  
I = 2  
J = 2  
HULP = 0  
GO TO 1000

C  
C COMPRIMATIE VAN KLINKERS

C  
101 IF(LETTER .LE. 25 .OR. LETTER .GE. 41) GO TO 102  
VOLG LET = WOORD(I + 1)  
IF(VOLG LET .LE. 28 .OR. VOLG LET .GE. 43) GO TO 102  
LETTER = COMPR(LETTER - 25, VOLG LET - 28)  
IF(LETTER .LE. 0) GO TO 102  
I = I + 1  
HULP = HULP + 1

```

        WOORD(I) = LETTER
        GO TO 1000
102  WOORD(J) = WOORD(I)
        J = J + 1
        I = I + 1
        VERW(J) = VERW(J) + HULP
C
C OPGEBOUWD
C
1000 IF(I .EQ. AANTAL) GO TO 1001
        LETTER = WOORD(I)
        GO TO 101
C
C BEPALING AANTAL SYLLABEN
C
1001 WOORD(J) = WOORD(AANTAL)
        AANTAL = J
        J = 1
        I = 2
201  IF(WOORD(I) .GE. 39) GO TO 202
        KLINKER(J) = I
        J = J + 1
202  I = I + 1
        IF(I .LE. AANTAL .AND. J .LE. 20) GO TO 201
        AANT SYL = J - 1
        IF(AANT SYL .LT. 2) GO TO 9999
        BEURT I = 1
        VERSCH = 1
        START 1 = 1
        START 2 = AANT SYL
C
C WEER
C
301  I = AANT SYL
302  IF(KLINKER(I) - KLINKER(I - 1) .NE. VERSCH) GO TO 303
        BEURT(BEURT I) = I - 1
        BEURT I = BEURT I + 1
303  I = I - 1
        IF(I .GE. 2) GO TO 302
        IF(VERSCH .EQ. 1) START 1 = BEURT I
        IF(VERSCH .EQ. 2) START 2 = BEURT I
        IF(BEURT I .GE. AANT SYL) GO TO 304
        VERSCH = VERSCH + 1
        GO TO 301
304  BEURT I = START 2 - 1
        STOP = AANT SYL
        NIVO = 1
C
C VOLGENDE
C
401  BEURT I = BEURT I + 1

```

```

        IF(BEURT 1 .LT. STOP) GO TO 405
        IF(NIVO - 2) 402, 403, 9999
402 BEURT 1 = START 2
        GO TO 404
403 BEURT 1 = START 1
        STOP = START 2
404 NIVO = NIVO + 1
        IF(BEURT 1 .GE. STOP) GO TO 401
405 HULP = BEURT(BEURT 1)
        KLINK 1 = KLINKER(HULP)
        KLINK 2 = KLINKER(HULP + 1)
        IF(NIVO - 2) 501, 601, 701
501 I = KLINK 2 - 2
502 HULP = WOORD(I) - 38
        J = WOORD(I + 1) - 38
        HULP = CONS(HULP, J)
        SPLITS = I
        IF(IABS(HULP) .EQ. 1) GO TO 7777
        IF(HULP .LT. 0) GO TO 401
        I = I - 1
        IF(I .GE. KLINK 1 + 1) 502, 401
C
C EINDE EERSTE FASE
C
601 I = KLINK 2 - 2
602 J = WOORD(I) - 38
        HULP = WOORD(I + 1) - 38
        HULP = CONS(J, HULP)
        SPLITS = I
        IF(IABS(HULP) - 3) 7777, 603, 604
603 SPLITS = I - 1
        IF(I .EQ. KLINK 1 + 1) GO TO 7777
        HULP = WOORD(I - 1) - 38
        HULP = IABS(CONS(HULP, J))
        IF(HULP .EQ. 2) GO TO 7777
        SPLITS = I - 2
        IF(HULP .EQ. 3) GO TO 7777
        GO TO 605
604 IF(IABS(HULP) .EQ. 4 .AND. I .EQ. KLINK 2 - 2) GO TO 7777
605 IF(HULP .LT. 0) GO TO 401
        I = I - 1
        IF(I .GE. KLINK 1 + 1) 602, 401
C
C EINDE TWEEDE FASE
C
701 HULP = WOORD(KLINK 1 + 1)

```



```

SPLITS = KLINK 1
IF(HULP .EQ. LETT J .OR.
1  HULP .EQ. LETT V .OR.
2  HULP .EQ. LETT Z) GO TO 7777
IF(HULP .EQ. LETT X) GO TO 401
HULP = WOORD(KLINK 1)
SPLITS = KLINK 1 + 1
IF(HULP .EQ. LETT EE) GO TO 401
IF(HULP .GT. 24 .AND. HULP .LT. 29) GO TO 7777
HULP = 4096 × WOORD(KLINK 1 - 1) + 64 × HULP + WOORD(KLINK 1 + 1)
SPLITS = KLINK 1
IF(HULP .EQ. VEN .OR.
1  HULP .EQ. TER .OR.
2  HULP .EQ. NOG .OR.
3  HULP .EQ. DER .OR.
4  HULP .EQ. VER .OR.
5  HULP .EQ. HIER .OR.
6  HULP .EQ. VAN) 401, 7777

C
C EINDE DERDE FASE
C
C SPLITS AF NA
C
7777 BESTESP = VERW(SPLITS + 1) + EERSTE - 2
IF(BESTESP .LT. LAATSTE - 1 .AND. BESTESP .GT. EERSTE) RETURN
GO TO 401

C
C GEEN SPLITSING
C
9999 BESTESP = EERSTE - 1
RETURN
END

```