

STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM
REKENAFDELING

Automatisch scheiden van Duitse lettergrepen

door

H. Brandt Corstius

en

E.G.M. Broerse

NR 2



september 1967

§ 1 Inleiding

Dit rapport behandelt de automatische splitsing van Duitse woorden in spellingslettergrepen. Met typografische eisen als: 'niet één enkele letter van een woord afsplitsen' werd geen rekening gehouden. Evenmin werd gezocht naar de meest geschikte splitsing van een woord in twee delen. Voor elk woord wordt één splitsing gegeven, zodat homografen als 'Wachtraum' slechts in één betekenis juist gesplitst worden.

De methode die bij het in § 3 afgedrukte ALGOL-programma werd toegepast is grotendeels dezelfde als die van een programma dat Nederlandse woorden in lettergrepen splitst en dat gepubliceerd werd in [1]. Een andere vulling van de lijsten met voorvoegsels, achtervoegsels, uitspreekbare medeklinkercombinaties en klinkers bleek echter nog niet geheel voldoende, zodat enkele statements in het programma werden opgenomen speciaal voor de Duitse taal.

Als autoriteit op het gebied van de Duitse spelling namen we Duden [2]. Een woordenboek waarin alle woorden gesplitst staan opgegeven hebben we niet gevonden. De beste publicatie op het gebied van de Duitse spellingslettergreep is die van Geisli [3]. Martin en O'Brien hebben in [4] hun programma beschreven, zonder echter voor ons nuttige details te geven. Wel is vermeldenswaard dat zij, afhankelijk van het soort materiaal, 0,7 tot 2% fouten maken, en dat de 'main factor in this variation seems to be the proportion of compound words in the text: at least 60% of errors occur at the juncture point of a compound word'.

In § 2 beschrijven we de door ons gekozen oplossing en geven we, in letters, de vulling van de arrays Comprimatie, Tweecons, Voorvoegsel, en Achtervoegsel. § 3 bevat de tekst van het ALGOL programma en de numerieke vulling van de arrays. § 4 bespreekt de resultaten van de toepassing van het programma op enkele lijsten Duitse woorden om de mate van succes te meten.

Een belangrijk element van ons programma is dat het geen uitzonderingslijsten kent, die, in verband met de veel voorkomende samenstellingen, niet nuttig zouden zijn.

§ 2 Beschrijving van het programma

2.1 Vulling van de lijsten

Het array comprimatíe zorgt voor het comprimeren van

aa, ai, au, ay,

ee, ei, eu, ey,

ie,

oa, oi, oo, ou, oy,

ui,

ch,

sz,

qu.

In de achtervoegsel-lijst (array AV) zijn opgenomen:

3 letters	4 letters	5 letters
heit	ling	angst
keit	arzt	abend
lich	land	
lein		
los		
tum		
lig		
ler		
weis		
tag		

In de voorvoegsel-lijst (array VV) zijn opgenomen:

2 letters	3 letters	4 letters
er	ver	durch
an	vor	fort
aus	ent	ruck
un	dar	bund
ein	her	
ab	hin	
auf	nach	
in	emp	
um	hier	
ur	zer	
ex	mit	
ob	unt	
ub	int	
	nied	
	geg	
	leb	
	ost	
	wied	

In de meercons-lijst (array MEERCONS) zijn opgenomen:

2 letters	3 letters
bl ph	pfl
br pl	schl
dr pr	schm
fl sch	schn
fr sp	schr
gl st	schw
gr tr	spr
kl zw	str
kn chr	
kr	

2.2. Speciale Maatregelen

2.2.1. Toepassing van de lijsten

Voor een maximaal rendement der lijsten bleken de volgende maatregelen noodzakelijk.

Bij de comprimatie

- ee: twee opeenvolgende e's werden gecomprimeerd, tenzij
- voorafgegaan door een g (ge-endet, ge-eignet).
 - voorafgegaan door een b én gevolgd door nog minstens 4 letters (be-eilen, Erdbee-ren)

N.B. sz: wordt altijd gecomprimeerd, dit i.v.m. de scherpe s, dus au-szer, maar helaas ook au-sziehen. Bij gebruik van een enkel teken (β) voor sz doet dit probleem zich niet voor.

Bij achtervoegsels

Teneinde niet alle achtervoegsels met hun mogelijke verbuigingen te hoeven opnemen worden de wordeinden allereerst onderzocht op de uitgangen: e, s, es, er, em, en.

Er wordt iteratief naar achtervoegsels gezocht (Wirk-lich-keit).

In verband hiermee worden los en ler als achtervoegsel apart behandeld.

Bij voorvoegsels

- meerlettergrepige voorvoegsels. Deze worden zelf ook gesplitst en dienen zowel in de lijst als in het programma apart behandeld te worden.

Hieronder vallen:

- unt - unter
- int - inter
- nied - nieder
- geg - gegen
- ub - uber
- ob - ober
- ab - aber
- leb - lebens

- b. voorvoegsels die soms niet afgesplitst moeten worden.
her: niet afgesplitst indien gevolgd door z, b, i of d.
(Herz, Herbst, Hering, Herd enz.)
mit: indien gevolgd door tw, dan splitsing mitt-w

2.2.2. Vermoede Samenstellingen

Indien de lijsten geen splitsing opleveren, wordt er gesplitst volgens

- a. V - V of
b. VC(C) - CV.

Van geval a wordt afgeweken indien VV=io en indien VV=yo of VV=ya aan het begin van een woord.

Van geval b wordt afgeweken indien

$$\text{CVC} = \left\{ \begin{array}{c} \text{s} \\ \text{n} \end{array} \right\} \text{ a } \left\{ \begin{array}{c} \text{n} \\ \text{b} \\ \text{r} \\ \text{l} \end{array} \right\}$$

en tevens

de a door minstens 4 letters wordt gevolgd (achtervoegsels daargelaten) en door minstens 4 letters wordt voorafgegaan.

Er wordt dan gesplitst volgens VC(C)C - V (Kindes-alter, Versuchsanlage, Wirtschafts-abkommen, Spitzen-arbeit, Kosten-anschlag, Ge-sang).

2.2.3. Verandering van letters

ck: Bij een splitsing tussen c en k, wordt de c een k.
(zucker → zuk-ker).

N.B. niet verwerkt werd het geval van de 3 gelijke medeklinkers
(Schiffahrt → Schiff-fahrt).

2.3. Numerieke vulling der lijsten en de interne code

Alle in lijsten opgenomen woordgedeelten dienen eerst gecomprimeerd te worden alvorens de numerieke waarde ervan te bepalen; er geldt dan:

voorvoegsel abcde = (((a × 50 + b) × 50 + c) × 50 + d) × 50 + e
achtervoegsel abcde = (((e × 50 + d) × 50 + c) × 50 + b) × 50 + a
meercons abc = (c × 50 + b) × 50 + a

de interne code is vastgelegd door het array STANDAARD en luidt:

ingevoerd	codering
symbool	
a	1
.	.
.	.
.	.
.	.
.	.
z	26
-	27 onderstreping
	28 verticale doorstreping
.	29 punt
,	30 apostrof

Niet in deze lijst voorkomende symbolen worden overgeslagen, uitgezonderd de woordscheiders en de tekstafsluiter.

Voor de interne code van de gecomprimeerde letters, zie de procedure vul.

2.4. Aanbrengen van wijzigingen in de lijsten.

Het aantal interne symbolen kan in de huidige opzet de 49 niet overschrijden. Hiermee dient men rekening te houden bij het veranderen van de comprimatielijst, die m.b.v. de procedure vul gewijzigd kan worden.

Voor het toevoegen van voor- en achtervoegsels dient men de resp. de SWVV en SWLL aan te passen en meestal ook een stuk programma.

Meercons kan men zonder meer uitbreiden.

§ 3 ALGOL-programma en vulling van de arrays

```
begin integer k, n, sym, koppelteller;
  integer array STANDAARD[0:127], W[1:60], comprimatie[0:48,0:30],
  koppel[0:25];
  array AV, VV, MEERCONS[2:5,0:30];

  procedure nextsymbol;
  begin switch SW:= NIETTOEGELATEN, WOORDSCHEIDER, TEKSTAFSLUITER;
  NIETTOEGELATEN: sym:= STANDAARD[RESYM];
    goto if sym < 0 then SW[ - sym] else WOORDEENHEID;
  WOORDSCHEIDER:
  end;

  procedure drukaf(element); value element; integer element;
  if element ≠ 0 then PRSYM(if element < 27 then element + 9 else
  if element < 29 then element + 99 else if element = 29 then 88
  else 120);

  procedure vul;
  begin integer k, j;
    real aantal;
    for k:= 0 step 1 until 127 do STANDAARD[k]:= read;
    for k:= 2 step 1 until 5 do
    begin aantal:= AV[k,0]:= read;
      for j:= 1 step 1 until aantal do AV[k,j]:= read
    end;
    for k:= 2 step 1 until 5 do
    begin aantal:= VV[k,0]:= read;
      for j:= 1 step 1 until aantal do VV[k,j]:= read
    end;
    for k:= 2, 3 do
    begin aantal:= MEERCONS[k,0]:= read;
      for j:= 1 step 1 until aantal do MEERCONS[k,j]:= read
    end;
```



```
for k:= 1 step 1 until 48 do  
for j:= 1 step 1 until 30 do comprimatie[k,j]:= - j;  
for k:= 1 step 1 until 48 do comprimatie[k,0]:= - 100;  
for k:= 1 step 1 until 27 do comprimatie[27,k]:= k;  
comprimatie[1,1]:= 31; comprimatie[1,21]:= 32;  
comprimatie[1,9]:= 33; comprimatie[1,25]:= 34;  
comprimatie[5,5]:= 35; comprimatie[5,9]:= 36;  
comprimatie[5,21]:= 37; comprimatie[5,25]:= 38;  
comprimatie[9,5]:= 39; comprimatie[15,9]:= 41;  
comprimatie[15,15]:= 42; comprimatie[15,21]:= 43;  
comprimatie[15,25]:= 44; comprimatie[21,9]:= 45;  
comprimatie[17,21]:= 48; comprimatie[19,26]:= 47;  
comprimatie[3,8]:= 46  
end;
```

```
procedure splits(aantalletters); value aantalletters;  
integer aantalletters;  
begin integer a, e, i, o, u, y, n, g, b, k, s, t, d, c, z, l, r,  
    ee, ub, letter, volgendeletter, eersteklinker, tweedeklinker,  
    woordbegin, wordeind, ei, tw, koppelteller, avkoppelteller,  
    aantalcons, meercons, langsteachvgs1, uitgang, entry, t1, t2,  
    pseudowordeind;  
real av, vv;  
integer array VERWIJZING, w[0:50], avkoppel[1:5];  
array SLOT, VOOR[2:5];  
switch SWVV:= SPL, SPL, SPL, SPL, SPL, AB, SPL, SPL, SPL,  
SPL, OB, UB, SPL, SPL, SPL, SPL, SPL, SPL, SPL, SPL, SPL,  
SPL, HER, SPL, SPL, SPL, SPL, SPL, MIT, UNT, INT, NIED, GEG,  
LEB, WIED, SPL, SPL, SPL, SPL, SPL, SPL, BUND, SPL;  
switch SWLL:= RW1, RW1, RW1, D, E, RW1, G, RW1, RW1, RW1, RW1,  
RW1, M, N, RW1, RW1, RW1, R, S, T, RW1, RW1, RW1, RW1, RW1,  
RW1, RW1, RW1, RW1, RW1, RW1, RW1, RW1, RW1, RW1, RW1, RW1,  
RW1, RW1, RW1, RW1, RW1, RW1, RW1, RW1, CH, RW1;
```

```
procedure splitsaf(letternr, label); value letternr;
integer letternr; label label;
begin koppelteller:= koppelteller + 1;
    koppel[koppelteller]:= VERWIJZING[letternr];
    woordbegin:= letternr + 1; goto label
end;

boolean procedure klinkers;
begin klinkers:= false; eersteklinker:= tweedeklinker;
    for t1:= eersteklinker + 1 step 1 until wordeind do
    begin letter:= w[t1];
        if (letter > 30  $\wedge$  letter < 46)  $\vee$  letter = a  $\vee$  letter = e  $\vee$ 
        letter = i  $\vee$  letter = o  $\vee$  letter = u  $\vee$  letter = y then
            begin klinkers:= true; tweedeklinker:= t1; goto end end
        end;
    end;
end;
end;

INIT: t1:= t2:= koppelteller:= W[aantalletters + 1]:= 0;
    avkoppelteller:= 6; letter:= W[1]; woordbegin:= a:= 1; e:= 5;
    t:= 20; i:= 9; o:= 15; u:= 21; y:= 25; ee:= 35; ei:= 39; n:= 14;
    r:= 18; g:= 7; b:= 2; k:= 11; c:= 3; ub:= 13; s:= 19; d:= 4;
    z:= 26; l:= 12; tw:= 23;

COMPR: t1:= t1 + 1;
COMPR1: t2:= t2 + 1;
    volgendeletter:= comprimatie[letter,W[t2 + 1]];
    if volgendeletter > 0 then
        begin if volgendeletter = ee then
            begin if w[t1 - 1] = g  $\vee$  (w[t1 - 1] = b  $\wedge$  n - t2 > 3) then
                begin volgendeletter:= - e; goto VERWIJS end
            end;
            letter:= volgendeletter; goto COMPRI
        end
    else
```

```
begin
VERWIJS: w[t1]:= letter; VERWIJZING[t1]:= t2;
  letter:= - volgendeletter;
  if t2 < aantalletters then goto COMPR
end;
woordeind:= t1; w[t1 + 1]:= 0;
ACHVGS: tweedeklinker:= 0; klinkers; if  $\neg$ klinkers then
  begin if eersteklinker = 0 then avkoppelteller:= avkoppelteller
    + 1; goto AFWERKING
  end;
  goto SWLL[w[woordeind]];
E: uitgang:= 1; goto ZOEKACHVGS;
S: uitgang:= if w[woordeind - 1] = e then 2 else if w[woordeind
  - 1] = o  $\vee$  w[woordeind - 1] = ei then 0 else 1; goto ZOEKACHVGS;
R: if w[woordeind - 1] = e then
  begin uitgang:= if w[woordeind - 2] = 1 then 0 else 2;
    goto ZOEKACHVGS
  end
  else goto RW1;
M:
N: if w[woordeind - 1] = e then
  begin uitgang:= 2; goto ZOEKACHVGS end;
D:
T:
G:
CH: uitgang:= 0;
ZOEKACHVGS: pseudowoordeind:= woordeind - uitgang;
  langsteachvgs:= if pseudowoordeind > 5 then 5 else
  pseudowoordeind; av:= w[pseudowoordeind];
  for t1:= 2 step 1 until langsteachvgs do SLOT[t1]:= av:= av  $\times$  50
  + w[pseudowoordeind - t1 + 1];
  for t1:= langsteachvgs step - 1 until 2 do
  begin av:= SLOT[t1];
    for t2:= AV[t1,0] step - 1 until 1 do if av = AV[t1,t2] then
    begin if w[pseudowoordeind + 1] = e then
```

```
begin avkoppelteller:= avkoppelteller - 1;
  avkoppel[avkoppelteller]:= VERWIJZING[pseudowordeind - 1]
end;
woordeind:= pseudowordeind - t1;
avkoppelteller:= avkoppelteller - 1;
avkoppel[avkoppelteller]:= VERWIJZING[woordeind];
goto ACHVGSL
end
end;
goto RW1;
RESTWOORD: if  $\neg$ klinkers then goto AFWERKING;
RW1: aantalcons:= tweedeklinker - eersteklinker - 1;
if aantalcons = 0 then
  begin if (w[eersteklinker] = i  $\wedge$  w[tweedeklinker] = o)  $\vee$ 
    (w[eersteklinker] = y  $\wedge$  w[tweedeklinker] = o)  $\vee$ 
    (w[eersteklinker] = y  $\wedge$  w[tweedeklinker] = a) then goto
    RESTWOORD else splitsaf(eersteklinker, RESTWOORD)
  end;
VOORVGS: t1:= tweedeklinker - woordbegin; if t1 < 6 then
  begin vv:= w[woordbegin];
  for t2:= 2 step 1 until t1 do VOOR[t2]:= vv:= vv  $\times$  50 +
  w[woordbegin + t2 - 1];
  for t1:= t1 step - 1 until 2 do
  begin vv:= VOOR[t1];
  for t2:= VV[t1,0] step - 1 until 1 do if vv = VV[t1,t2] then
  begin entry:= (t1 - 2)  $\times$  20 + t2;
  goto SWVV[if entry > 45 then 45 else entry];
  HER: volgendeletter:= w[woordbegin + 3];
  goto if volgendeletter = b  $\vee$  volgendeletter = i  $\vee$ 
  volgendeletter = r  $\vee$  volgendeletter = z  $\vee$  volgendeletter
  = d then TWEEOFDRIECONS else SPL;
  UNT:
  INT:
  NIED:
  WIED: if w[woordbegin + 3] = e  $\wedge$  w[woordbegin + 4] = r
```

```
then
begin
MEERLETTGRVVGSL: splitsaf(tweedeklinker - 2, VERVOLGVV);
VERVOLGVV: if klinkers then splitsaf(eersteklinker + 1,
      RESTWOORD) else goto AFWERKING
end
else goto ZOEKVVGSL;
BUND: goto if w[woordbegin + 4] = e  $\wedge$  w[woordbegin + 5] =
      s then MEERLETTGRVVGSL else ZOEKVVGSL;
MIT: if w[woordbegin + 3] = t  $\wedge$  w[woordbegin + 4] = tw
      then splitsaf(eersteklinker + 2, RESTWOORD) else goto
      SPL;
GEG: if w[woordbegin + 3] = e  $\wedge$  w[woordbegin + 4] = n
      then goto MEERLETTGRVVGSL else goto ZOEKVVGSL;
OB:
UB: goto if w[woordbegin + 2] = e  $\wedge$  w[woordbegin + 3] = r
      then MEERLETTGRVVGSL else ZOEKVVGSL;
AB: goto if w[woordbegin + 2] = e  $\wedge$  w[woordbegin + 3] = r
      then MEERLETTGRVVGSL else if w[woordbegin + 2] = e  $\wedge$ 
      w[woordbegin + 3] = n  $\wedge$  w[woordbegin + 4] = d then MLGVV
      else SPL;
LEB: if w[woordbegin + 3] = e  $\wedge$  w[woordbegin + 4] = n  $\wedge$ 
      w[woordbegin + 5] = s then
      begin
      MLGVV: splitsaf(eersteklinker, RESTMLGVV);
      RESTMLGVV: if klinkers then splitsaf(eersteklinker + 2,
        RESTWOORD) else goto AFWERKING
      end;
      SPL: splitsaf(woordbegin + t1 - 1, RESTWOORD)
      end
      else
      ZOEKVVGSL:
      end
end;
```

```
TWEEOFDRIECONS: for t1:= 3, 2 do if t1 < aantalcons then
  begin meercons:= if t1 = 3 then w[tweedeklinker - 1] × 2500 +
    w[tweedeklinker - 2] × 50 + w[tweedeklinker - 3] else
    w[tweedeklinker - 1] × 50 + w[tweedeklinker - 2];
  for t2:= MEERCONS[t1,0] step - 1 until 1 do if meercons =
    MEERCONS[t1,t2] then splitsaf(tweedeklinker - t1 - 1,
    RESTWOORD)
  end;
  if w[tweedeklinker] = a ∧ (w[tweedeklinker + 1] = n ∨
  w[tweedeklinker + 1] = b ∨ w[tweedeklinker + 1] = r ∨
  w[tweedeklinker + 1] = 1) ∧ (w[tweedeklinker - 1] = s ∨
  w[tweedeklinker - 1] = n) ∧ (woordeind - tweedeklinker ≥ 4) ∧
  tweedeklinker > 4 then splitsaf(tweedeklinker - 1, RESTWOORD);
  if w[tweedeklinker - 1] = k ∧ w[tweedeklinker - 2] = c then
  W[VERWIJZING[tweedeklinker - 2]]:= k;
  splitsaf(tweedeklinker - 2, RESTWOORD);
AFWERKING: for t1:= avkoppelteller step 1 until 5 do
  begin koppelteller:= koppelteller + 1;
  koppel[koppelteller]:= avkoppel[t1]
  end;
  koppel[0]:= koppelteller; koppel[koppelteller + 1]:= - 100
  end;
INIT: vul;
VOORWOORD: n:= 0;
VOORWOORD1: nextsymbol; goto VOORWOORD1;
WOORDEENHEID: n:= n + 1; W[n]:= sym;
RESTWOORD: nextsymbol; W[n + 1]:= 0; splits(n); koppelteller:= 1;
NLCR;
for k:= 1 step 1 until n do
  begin drukaf(W[k]); if k = koppelteller then
    begin PRSYM(65); koppelteller:= koppelteller + 1 end
  end;
  goto VOORWOORD;
TEKSTAFSLUITER:
end
```

'het array STANDAARD:'

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	-1	1	2	3
4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23
24	25	26	-1	-1	-2	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	29	-1
-1	-1	-1	-2	-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	-1	-1	-1	-1	-1	-1	-2	-2
30	-1	-3	-1	-1	-1	27	28		

'achtervoegsels'

0	'achtervoegsels van twee letters'			
10	'achtervoegsels van drie letters'			
51808	51811	115462	36812	48262
33570	17962	45262	49323	17570
3	'achtervoegsels van vier letters'			
910462	2565901	535062		
2	'achtervoegsels van vijf letters'			
127393201	26762601			

'voorvoegsels'

13	'voorvoegsels van twee letters'			
268	64	1619	1064	1814
52	1606	464	1063	1068
274	752	1052		
18	'voorvoegsels van drie letters'			
55268	55768	13220	10068	20268
20464	35096	13166	21968	65268
32970	53220	23220	36954	17757
30252	38470	59454		
4	'voorvoegsels van vier letters'			
553446	788420	2302661	303204	

'meercons'

0				
19	'elementen van twee letters'			
602	902	904	606	906
607	907	611	711	911
416	616	916	2319	819
1019	920	1176	946	
8	'elementen van drie letters'			
30316	32319	34819	37319	47319
59819	45819	46019		

§ 4 Resultaten

Helaas stond ons geen modern frequentiewoordenboek van het Duits ter beschikking. Dat van Kaeding [5], dat we raadpleegden, is tezeer verouderd. Het programma van §3 werd toegepast op de 600 woorden bovenaan de bladzijden van een klein Duits-Nederlands woordenboek [6]. Van deze 600 woorden werden er 18, ofwel 3%, foutief gesplitst. Deze 3% dient men goed te onderscheiden van het foutenpercentage op de woorden van een Duitse tekst, daar hierbij de frequentie van de woorden een grote rol gaat spelen. Om een inzicht te krijgen in dit foutenpercentage werd het programma toe gepast op 400 woorden uit een aantal Duitse kranten [7] die aan het eind van de regel waren gesplitst. Nu werd dus alleen gecontroleerd of deze aangetroffen splitsing ook door het programma werd voortgebracht. Dit bleek voor 3 woorden, ofwel 1%, niet het geval te zijn. Het eigenlijke foutenpercentage van ons programma ligt waarschijnlijk nog lager omdat de menselijke splitsing een voorkeur heeft voor de splitsing tussen de delen van een samenstelling, die voor het programma juist moeilijkheden oplevert. Het foutenpercentage lijkt aanvaardbaar voor de toepassing bij het automatisch zetten van Duitse tekst.

- [1] H. Brandt Corstius, Automatisch tellen en scheiden van Nederlands lettergrepen, Mathematisch Centrum, MR 67, 1964.
- [2] Duden, deel 7: Rechtschreibung, 15e druk
- [3] J.A. Geissl, Worttrennung in Deutsch und English, Der Druckspiegel 19(1964), 1-28.
- [4] M.D. Martin and E.J. O'Brien, An approach to the automatic syllabification of German, Proc. IFIP Congress '65, vol. 2 p. 491.
- [5] F.W. Kaeding, Häufigkeitwörterbuch der Deutschen Sprache, Berlin 1898.
- [6] Prisma Woordenboek Duits-Nederlands, Het Spectrum, Utrecht, derde druk.
- [7] Frankfurter Allgemeine, Neues Deutschland, Der neue Tag, Süddeutsche Zeitung, Frankfurter Rundschau, Saarbrücker Zeitung. Alle van 31 augustus 1967.

Summary

A program for the automatic division into spelling syllables of German words is given in ALGOL 60. No exception list is used.

This program fails for 3% of the word types in a sample of a dictionary and for 1% of the word tokens in German printed material as they were actually hyphenated by the typesetter. The error percentage is acceptable for application in automatic typesetting of German text.

Zusammenfassung

Es wird ein ALGOL 60 - Programm für die automatische Einteilung in geschriebene Silben von deutschen Wörtern gegeben. Eine Ausnahmenliste wird dabei nicht verwendet.

Es werden Fehler gemacht bei 3% der Worttypen einer Auswahl aus einem Wörterbuch, und bei 1% der Wortzeichen in deutschen gedruckten Texten wie sie tatsächlich von dem Setzer geteilt wurden. Die wirkliche Fehlerprozentsatz ist weniger als 1% und damit akzeptabel bei Anwendung in automatischem Setzen.