

STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

REKENAFDELING

Automatisch scheiden van Amerikaans-Engelse lettergrepen

door

H. Brandt Corstius

en

E.G.M. Broerse

NR 6



juni 1968

RA

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.

1 Inleiding

Dit rapport behandelt de automatische splitsing van Engelse woorden in spellingslettergrepen, zoals dat in de Verenigde Staten gebruikelijk is. Duncan [1] vond een aanzienlijk verschil tussen de Britse en Amerikaanse voorschriften voor lettergreepsplitsing.

Voor elk woord wordt slechts één splitsing aangegeven, zodat (niet-zeldzame) homografen als "adder" slechts in één betekenis juist gesplitst worden.

Als autoriteit op het gebied van de Amerikaanse spelling beschouwden we Webster [2], dat van alle woorden de lettergreepsplitsing opgeeft. De Engelse lettergreepsplitsing is zo grillig dat men in de praktijk vaak de oplossing van het volledige woordenboek heeft gekozen [3].

Onze opgave was een programma dat minder tijd en ruimte zou kosten, maar dan ook niet volmaakt zou zijn. Gepubliceerde programma's hebben we niet gevonden, wel een - onbevredigend - algoritme [4] en een onvolledige beschrijving [5] van een programma dat minder dan 10% fouten zou maken [6]. Recent is de publikatie van Mostert [7].

In het algemeen zijn er twee benaderingen van het probleem: 1. de wijze waarop Nederlandse, Duitse en Zweedse woorden gesplitst werden [8,9,10] met behulp van lijsten van voorvoegsels, achtervoegsels en uitspreekbare medeklinker-combinaties. Deze aanpak bleek, ook na het programmeren van allerlei speciale maatregelen nog ongeveer 20% fouten te maken. 2. de statistische methode waarbij, op grond van een groot basismateriaal, de kans dat vóór of na een bepaald bigram van letters een lettergreepsplitsing voorkomt wordt gebruikt. Ook deze aanpak bleek ongeveer 20% fouten te maken. Wij verkozen ten slotte een vermenging van deze twee methoden, waarbij wel voor- en achtervoegsels gebruikt worden, maar de beslissing of een bepaalde medeklinker bij de vorige of de volgende lettergreep behoort op statistische gronden beslist wordt wanneer er minder dan 3 medeklinkers tussen de klinkers staan, terwijl anders de lijst van uitspreekbare medeklinkercombinaties de doorslag geeft.

De beschrijving van deze oplossing vindt men in §2. §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 Engelse woorden om de mate van succes te meten.

§2. Beschrijving van het programma

De "statistische" methode werkt als volgt:

We gaan er vanuit dat gesplitst wordt tussen 2 klinkers (er wordt niet gecomprimeerd).

We zeggen nu dat een splitsingsveld gevormd wordt door alle mogelijke (splitsings)plaatsen vanaf een eerste klinker tot aan de daaropvolgende klinker.

Vb. kopstand, splitsingsveld: opst ,
d.w.z. we hebben de splitsingsplaatsen:
ko-ps, op-st, ps-ta, st-an.

Aan alle splitsingsplaatsen binnen een splitsingsveld wordt een waarde toegekend en wel het produkt van de bigramwaarde voor de splitsingsplaats en de bigramwaarde erna.

De door ons gekozen splitsing is die waaraan de grootste waarde is toegekend.

Het bepalen van de bigramwaarden gebeurde m.b.v. woorden getrokken uit een frequentielijst [11].

Deze woorden werden gesplitst volgens Webster, in splitsingsvelden verdeeld, waarna binnen ieder splitsingsveld de waarde van het bigram voor de splitsingsplaats met de frequentie van het woord werd verhoogd, evenzo de waarde van het bigram na de splitsingsplaats.

Iedere bigram krijgt op deze manier 2 essentieel verschillende waarden toegewezen.

Het programma in §3 is een combinatie van de "statistische" methode en de "klassieke" methode van Nederlands, Duits en Zweeds:

1. Er wordt niet gecomprimeerd.
2. Afsplitsen van voorvoegsels en achtervoegsels.
3. Speciale maatregelen voor bepaalde verbuigingen.
4. De overblijvende romp van het woord wordt in splitsingsvelden verdeeld.

Indien een splitsingsveld meer dan 2 medeklinkers bevat, splitsen we gebruik makend van een lijst uitspreekbare medeklinkercombinaties, in het andere geval splitsen we volgens de "statistische" methode.

Achtervoegsels

Iteratief worden achter afgesplitst:

ic, ly,
ism, ize, ice, man, gic, ric, cal, out,
less, ness, cian, nian, cion, gion, tion, sion,
cial, sial, tial, self, ment,
thing.

Voorvoegsels

Van voren worden afgesplitst:

ab, ac, ad, at, ep, en, un,
can, cap, car, col, mis, per.

Uitspreekbare medeklinkercombinaties

br, cr, dr, fr, pr, tr, wr, gr,
bl, fl, cl, pl, gl,
ch, ph, th, wh,
kn, sc, st,
str, thr, sph.

Speciale maatregelen

Een "e" aan het eind van een woord wordt niet als klinker beschouwd (stomme e), behalve in de gevallen:

-dle, -ble, -gle, -tle, -fle, -ple, -cre.

Voor de uitgangen "est", "er" en "ing" geldt het volgende:

Indien voorafgegaan door een dubbele medeklinker dan
resp. C-Cest, C-Cer, C-Cing.

Indien voorafgegaan door dl, bl, gl, tl, fl, pl, cr dan
eveneens tussen de medeklinkers splitsen.

"ed" aan het eind van een woord wordt als medeklinker beschouwd, behalve wanneer voorafgegaan door een dubbele medeklinker (C-Ced) of door een "t" of "d" (t-ed, d-ed).

Woorden eindigend op een "s":

Afgesplitst wordt van achter ch-es, ss-es, -ces, -ses indien van toepassing.

In alle andere gevallen: verwijder de "s", aannemende dat het een meervoud betreft en indien het woord nu eindigt op "e", "er" of "ing" geef het dan de hiervoor beschreven behandeling.

§3. ALGOL-programma en vulling van de arrays.

pag. 5 t/m 17.

§4. Resultaten

Het programma uit §3 werd toegepast op 1450 woorden uit twee Amerikaanse kranten [12] die daar aan het eind van de regel afgebroken waren. In 220 gevallen, dus 15%, bleek het programma een andere afbreking te geven. Een enkele maal was dit verschil, doordat de Amerikaanse krant zich niet aan de Webster hield. Belangrijker is dat onze anders gesplitste woorden geen bijzonder vreemde woordbeelden te zien geven, waardoor het programma waarschijnlijk geschikt is voor toepassing bij het automatisch zetten.

```
begin integer a, e, i, o, u, y, b, c, d, f, g, h, j, k, l, m, n, p,  
q, r, s, t, v, w, x, z, kappa, gamma, dl, bl, kl, gl, tl, fl, pl,  
cr, K, J, L, clc2, begin, eind, eersteklinker, uitgang,  
klinkerteller, koppelteller, avkoppelteller, woordbegin,  
woordeind, WOORDEIND, tussenklinkerwaarde, value, max, splitter;  
integer array STANDAARD[0:127], W[0:51], PERCENT[0:1,1:28,1:28],  
KLINKER[0:25], avkoppel[0:6], koppel2[0:20], VERM[2:3],  
OUTPUT[1:26];  
array SLOT[2:5], AV[2:5,0:25], VV[2:3,0:25], VOOR[2:3],  
MEERCONS[2:3,0:20];  
switch SWLL:= VV1, E, VV1, VV1, VV1, Y, VV1, C, D, F, G, VV1, VV1,  
VV1, LA, M, N, VV1, VV1, R, S, T, VV1, VV1, VV1, VV1, VV1, VV1,  
VV1, VV1, VV1, VV1, VV1, VV1, VV1, VV1, VV1, VV1, VV1, VV1;  
  
procedure vul;  
begin integer k, j, l;  
  real aantal;  
  for k:= 0 step 1 until 127 do STANDAARD[k]:= read;  
  for k:= 2, 3, 4, 5 do  
    begin AV[k,0]:= aantal:= read;  
      for j:= 1 step 1 until aantal do AV[k,j]:= read  
    end;  
  for k:= 2, 3 do  
    begin VV[k,0]:= aantal:= read;  
      for j:= 1 step 1 until aantal do VV[k,j]:= read  
    end;  
  for k:= 1 step 1 until 26 do OUTPUT[k]:= read;  
  for k:= 2, 3 do  
    begin MEERCONS[k,0]:= aantal:= read;  
      for j:= 1 step 1 until aantal do MEERCONS[k,j]:= read  
    end;  
  tussenklinkerwaarde:= read;  
  for k:= 0, 1 do  
    for j:= 1 step 1 until 28 do
```

```
for l:= 1 step 1 until 28 do PERCENT[k,j,l]:= read  
end;
```

```
procedure leeswoord;  
begin integer I, SYM;
```

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

```
    klinkerteller:= I:= 0; woordbegin:= 1;  
NEXTSYM: nextsymbol; I:= I + 1; W[I]:= SYM; if SYM < 7 then  
    begin klinkerteller:= klinkerteller + 1;  
        KLINKER[klinkerteller]:= I  
    end;  
    goto NEXTSYM;  
WOORDSCHEIDER: WOORDEIND:= wordeind:= I  
end;
```

```
procedure zoekachtervoegsel;  
begin integer langste, i, j;  
    real av;  
    langste:= if wordeind > 5 then 5 else wordeind;  
    av:= W[wordeind];  
    for i:= 2 step 1 until langste do SLOT[i]:= av:= av × 50 +  
    W[wordeind - i + 1];  
    for i:= langste step - 1 until 2 do  
    begin av:= SLOT[i];  
        for j:= AV[i,0] step - 1 until 1 do if av = AV[i,j] then  
        begin avkoppelteller:= avkoppelteller - 1;  
        wordeind:= wordeind - i;
```

```
    avkoppel[avkoppelteller]:= wordeind; goto ACHTERAAN
  end
end
end zoek achtervoegsel;
```

```
procedure zoekvoorvoegsel;
begin integer i, j;
  real vv;
  if wordeind  $\neq$  1 then
    begin vv:= W[1]; i:= 1;
      for i:= i + 1 while wordeind > i  $\wedge$  i < 4 do
        begin vv:= vv + W[i]  $\times$  VERM[i];
        for j:= VV[i,0] step - 1 until 1 do if vv = VV[i,j] then
          begin koppelteller:= koppelteller + 1;
            woordbegin:= woordbegin + i;
            koppel2[koppelteller]:= woordbegin - 1; goto END
          end
        end
      end;
    END:
  end;
```

```
procedure legvast(uitgang); value uitgang; integer uitgang;
begin avkoppelteller:= avkoppelteller - 1;
  wordeind:= wordeind - uitgang;
  avkoppel[avkoppelteller]:= wordeind
end leg vast;
```

```
INIT: VERM[2]:= 50; VERM[3]:= 2500; W[0]:= 27; KLINKER[0]:= - 1;
vul; a:= 1; e:= 2; i:= 3; o:= 4; u:= 5; y:= 6; b:= 7; c:= 8; d:= 9;
f:= 10; g:= 11; h:= 12; j:= 13; k:= 14; l:= 15; m:= 16; n:= 17;
p:= 18; q:= 19; r:= 20; s:= 21; t:= 22; v:= 23; w:= 24; x:= 25;
```

```

z:= 26; kappa:= 27; gamma:= 28; dl:= 759; bl:= 757; kl:= 764;
gl:= 761; tl:= 772; fl:= 760; pl:= 768; cr:= 1008;
VOLGENDEWOORD: leeswoord; avkoppelteller:= 7; koppelteller:= 0;
  if woordeind = 0 then goto VOLGENDEWOORD;
  if klinkerteller < 2 then goto AFWERKING; W[woordeind + 1]:= gamma;
ACHTERAAN: goto SWLL[W[woordeind]];
D: if W[woordeind - 1] = e then
  begin if W[woordeind - 2] = W[woordeind - 3] then legvast(3) else
    if W[woordeind - 2] = t  $\vee$  W[woordeind - 2] = d then legvast(2)
    else woordeind:= woordeind - 2; goto PREPVV
  end;
  goto ACHTERVGSL;
R: if W[woordeind - 1] = e then
  begin if W[woordeind - 2] = d  $\vee$  W[woordeind - 2] = t then
    legvast(3) else
    begin uitgang:= 2; goto RTG end
  end;
  goto PREPVV;
T: if W[woordeind - 2] = e  $\wedge$  W[woordeind - 1] = s then
  begin uitgang:= 3; goto RTG end;
  goto ACHTERVGSL;
G: zoekachtervoegsel;
  if  $\neg$ (W[woordeind - 2] = i  $\wedge$  W[woordeind - 1] = n) then goto
  PREPVV; uitgang:= 3;
RTG: if W[woordeind - uitgang] = W[woordeind - uitgang - 1] then
  legvast(uitgang + 1) else
  begin c1c2:= W[woordeind - uitgang]  $\times$  50 + W[woordeind - uitgang -
  1];
  if c1c2 = dl  $\vee$  c1c2 = bl  $\vee$  c1c2 = gl  $\vee$  c1c2 = tl  $\vee$  c1c2 = fl  $\vee$ 
  c1c2 = pl  $\vee$  c1c2 = cr then uitgang:= uitgang + 1;
  if uitgang  $\neq$  2 then legvast(uitgang)
  end;
  goto PREPVV;
E: zoekachtervoegsel;
  c1c2:= W[woordeind - 2] + W[woordeind - 1]  $\times$  50;

```

```
if c1c2 = d1 ∨ c1c2 = b1 ∨ c1c2 = g1 ∨ c1c2 = t1 ∨ c1c2 = f1 ∨  
c1c2 = pl ∨ c1c2 = cr then legvast(3) else wordeind:= wordeind  
- 1; goto PREPVV;  
S: if W[wordeind - 1] ≠ e then zoekachtervoegsel else  
begin if (W[wordeind - 3] = c ∧ W[wordeind - 2] = h) ∨  
(W[wordeind - 3] = s ∧ W[wordeind - 2] = s) then legvast(2)  
else if W[wordeind - 2] = c ∨ W[wordeind - 2] = s then  
legvast(3) else  
begin wordeind:= wordeind - 1; goto E end;  
goto PREPVV  
end;  
wordeind:= wordeind - 1;  
if W[wordeind - 1] = e ∧ W[wordeind] = r then  
begin uitgang:= 2; goto RTG end  
else if W[wordeind - 2] = i ∧ W[wordeind - 1] = n ∧ W[wordeind]  
= g then  
begin uitgang:= 3; goto RTG end  
else zoekachtervoegsel; goto PREPVV;  
ACHTERVGSL:  
C:  
LA:  
M:  
H:  
N:  
F:  
Y: zoekachtervoegsel;  
PREPVV:  
VV1: if KLINKER[klinkerteller] > wordeind then  
begin klinkerteller:= klinkerteller - 1; goto PREPVV end;  
if klinkerteller = 0 then avkoppelteller:= avkoppelteller + 1;  
if klinkerteller < 2 then goto AFWERKING; zoekvoorvoegsel;  
eersteklinker:= 100;  
for K:= 1 step 1 until klinkerteller do if KLINKER[K] ≥ woordbegin  
then
```

```
begin eersteklinker:= K; goto RESTWOORD end;  
RESTWOORD: if eersteklinker > klinkerteller then  
  begin koppelteller:= koppelteller - 1; goto AFWERKING end;  
RESTWOORD1: if eersteklinker = klinkerteller then goto AFWERKING;  
  begin:= KLINKER[eersteklinker];  
  eind:= KLINKER[eersteklinker + 1] - 1; max:= - 1;  
  if eind - begin > 2 then  
    begin splitser:= eind - 1; VOOR[2]:= W[eind] + W[eind - 1] × 50;  
      VOOR[3]:= VOOR[2] + W[eind - 2] × 2500;  
      for K:= 3, 2 do  
        begin J:= MEERCONS[K,0];  
          for L:= 1 step 1 until J do if VOOR[K] = MEERCONS[K,L] then  
            begin splitser:= eind - K; goto SPL end  
          end  
        end  
      else  
        begin for K:= begin step 1 until eind do  
          begin value:= (PERCENT[0,W[K - 1],W[K]] + 0.1) × (PERCENT[1,W[K +  
            1],W[K + 2]] + 0.1); if value > max then  
            begin splitser:= K; max:= value end  
          end  
        end;  
      end;  
    SPL: if splitser ≠ 1 ∧ splitser ≠ WOORDEIND - 1 ∧ ¬(begin = eind  
      ∧ max < tussenklinkerwaarde) then  
      begin koppelteller:= koppelteller + 1;  
        koppel2[koppelteller]:= splitser  
      end;  
      eersteklinker:= eersteklinker + 1; goto RESTWOORD1;  
    AFWERKING: for K:= avkoppelteller step 1 until 6 do  
      begin koppelteller:= koppelteller + 1;  
        koppel2[koppelteller]:= avkoppel[K]  
      end;  
      koppel2[koppelteller + 1]:= - 100; NLCR; koppelteller:= 1;  
      for K:= 1 step 1 until WOORDEIND do
```

```

begin PRSYM(OUTPUT[W[K]]); if K = koppel2[koppelteller] then
    begin PRSYM(65); koppelteller:= koppelteller + 1 end
end;
goto VOLGENDEWOORD;
TEKSTAFSLUITER:
end

```

het array STANDAARD:

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

achtervoegsels

achtervoegsels van 2 letters:

403 315

achtervoegsels van 3 letters:

41053 6303 5403 42566 20161 20170 37558 55254

achtervoegsels van 4 letters:

2677615	2677617	2127658	2127667	2135158
2135161	2135172	2135171	1877658	1877671
1877672	1287621	2792616		

achtervoegsels van 5 letters:

70883122

voorvoegsels

voorvoegsels van 2 letters:

351	401	451	1101	902	852	855
-----	-----	-----	------	-----	-----	-----

voorvoegsels van 3 letters:

42558	45058	50058	37708	50118	52666
-------	-------	-------	-------	-------	-------

het array OUTPUT:

10	14	18	24	30	34	11	12
13	15	16	17	19	20	21	22
23	25	26	27	28	29	31	32
33	35						

het array MEERCONS:

tweeletters:

370	420	470	520	920	1070	1220	365
515	412	912	1212	717	1058	1064	570
565	915	415	1112				

drieletters:

53620	55620	53412
-------	-------	-------

	0	0	0	0	0	0	1	0	0	0
		3	0	0	0	0	0	0	0	4261
b		12	33	32	80	4	12	0	0	0
	0	0	0	0	0	868	0	0	0	0
		0	0	0	0	0	0	0	0	0
c		129	58	180	178	76	20	0	0	0
	0	0	871	0	0	22	0	0	0	0
		20	0	0	0	0	0	0	0	0
d		110	970	188	1	22	16	0	0	0
	0	0	0	0	0	0	0	0	0	0
		49	0	0	0	0	0	0	0	0
f		10	99	43	10	867	20	0	0	0
	0	0	0	0	0	16	0	0	0	0
		6	0	0	0	0	0	0	0	0
g		28	22	109	3	28	12	0	0	0
	0	0	0	0	0	0	0	1	0	0
		834	0	0	0	0	0	0	0	0
h		5	19	2	1	0	4	0	0	0
	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0
j		0	1	0	0	2	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0
k		0	801	2	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
l		953	888	219	64	6	1268	0	0	0
	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0
m		922	1881	31	818	33	8	0	0	0
	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0
n		874	53	123	63	12	878	0	0	0
	0	0	0	0	0	0	0	0	0	0
		0	0	3	0	0	0	0	0	0

p	935	125	893	62	4	0	0	0	0
0	0	23	0	0	70	0	0	0	0
	104	0	0	0	0	0	0	0	0
q	0	0	0	0	23	0	0	0	0
0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
r	45	90	107	115	0	864	0	0	0
0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
s	822	943	205	32	48	7	0	22	0
0	0	87	0	1	0	0	0	40	0
	0	0	69	0	0	0	0	0	0
t	145	182	276	1066	104	366	0	0	0
0	0	115	0	0	801	0	0	0	0
	29	0	0	0	800	0	0	0	0
v	79	1077	24	847	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
w	79	0	2	2	0	0	0	0	0
0	0	70	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
x	0	2	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
z	45	0	6	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
K	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
G	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

K en G representeren de fictieve letter resp. voor en achter het woord.

- [1] J. Duncan, remarks Computer Typesetting Conference, London, 1965.
- [2] Webster's Seventh New Collegiate Dictionary, Merriam Company, Springfield Mass., 1965.
- [3] M.E. Stevens and J.L. Little, Automatic Typographic Quality Typesetting Techniques:
A State-of-the-Art Review, U.S. Gov. Printing Office, 1967.
- [4] R.P. Rich and A.G. Stone
Method for Hyphenating at the End of a Printed Line,
Comm. A.C.M. 8, 7, july '65.
- [5] J.L. Dolby, H.L. Resnikoff, L.W. Hart,
Automatic Syllabification of English words, proc. IEEE 51, 10,
oct. '63, p. 1371-1372.
- [6] Persoonlijke mededeling J.L. Dolby.
- [7] F.J.A. Mostert, The problem of Syllabification
Levende Talen, 241, oct. '67, 458-464.
- [8] H. Brandt Corstius, Automatisch tellen en scheiden van Nederlandse lettergrepen, Math. Centrum, MR 67, 1964.
- [9] H. Brandt Corstius en E.G.M. Broerse, Automatisch scheiden van Duitse lettergrepen, Math. Centrum, NR 2, 1967.
- [10] H. Brandt Corstius en E.G.M. Broerse, Automatisch scheiden van Zweedse lettergrepen, Math. Centrum, NR 5, 1968.
- [11] E.L. Thorndike and I. Lorge, The Teacher's Word Book, New York, 1944.
- [12] International Herald Tribune, 7 mei en 8 juni 1968.

Summary

A program for the automatic division into spelling syllables of words in American English is given in ALGOL 60. No exception list is used. The method is a mixture of the "statistical" and the "logical" approach.

The program fails for 15% of the word tokens in newspaper material as they were actually hyphenated by the typesetter. The nature of the mistakes seems to make the program acceptable for application in automatic typesetting of American-English text.

