

STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

REKENAFDELING

Automatisch scheiden van Zweedse lettergrepen

door

H. Brandt Corstius

en

E.G.M. Broerse

NR 5



mei 1968

§1. Inleiding

Dit rapport behandelt de automatische scheiding van Zweedse woorden in spellingslettergrepen. Voor elk woord wordt één splitsing gegeven, zodat (zeldzame) homografen als "bildrulle" 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 de programma's die Nederlandse en Duitse woorden splitsen [1,2]. Een andere vulling van de lijsten met voorvoegsels, achtervoegsels, uitspreekbare medeklinkercombinaties en klinkers bleek echter niet voldoende, zodat enkele statements in het programma werden opgenomen, speciaal voor de Zweedse taal.

De voorschriften voor de Zweedse lettergreepsplitsing vindt men, zij het niet voldoende exact, in [3]. Hoewel er meer dan één programma voor Zweedse splitsing bestaat, is er geen gepubliceerd.

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 het resultaat van de toepassing van het programma op een lijst van woorden uit Zweedse dagbladen om de mate van succes te meten.

2. Beschrijving van het programma

2.1. Comprimatie.

Tot klinkers worden gecomprimeerd de combinaties:

ai, au, ee, ei, ie, oi, oo, ou, ui.

Tot medeklinkers worden gecomprimeerd: ch en ng.

(Voor ng zie ook 2.4).

2.2. Achtervoegsels.

Iteratief worden achter afgesplitst:

ka, ra, ke, ta, ig,
 rat, lat, rad, nar, som, kar, rar,
 lek, sta, ska, dag, het, jet,
 land, slag, stad,
 grund, klart, aktig, strom.

2.3. Voorvoegsels.

Van voren worden afgesplitst:

be, ut, in, es, di,
 syd, for, lag, ost, lang, per, fol,
 dis, sam, ytt, dar, buk, sod, sak,
 tus, dig, upp, pro,
 verk, fort, form, dags, bets, folk,
 satt, siff, vagg, lags, lagg,
 tros, tags,
 bergs, freds, svars, mansk, skogs.

2.4. Twee- en driecons.

De volgende combinaties van 2 medeklinkers beginnen een lettergreep:

pr, tr, kr,
 bl, br, dr, sc, th, fl, fr, sv, sl, gj,
 sk, sch, sj, tj.

Ook behoren hiertoe: st (mits voorafgegaan door nog een medeklinker)
 en pl (mits voorafgegaan door een s).

De combinaties van 3 medeklinkers die een lettergreep beginnen zijn:
 spr, str, skr.

2.5. Speciale maatregelen.

Samenstellingen waarvan het eerste lid eindigt op ions- of ings-,
 of het tweede lid begint met -ut of -in (terwijl het eerste eindigt
 op een s) worden door enkele statements juist gesplitst.

Als de klinker voor ng de eerste letter van het woord is, wordt de
 comprimatie ongedaan gemaakt. Als op ng een klinker volgt wordt (in
 tegenstelling tot de algemene regel) het koppelteken na de ng gezet.
 Bij een x tussen klinkers wordt eveneens na de x gesplitst.

§3. ALGOL-programma en vulling van de arrays

begin integer k, n, sym, koppelteller;

integer array STANDAARD[0:127], W[0:60], comprimatie[0:32,0:32],

koppel2[0:25], MEERCONS[0:20];

array AV, VV[2:5,0:20];

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 $\neq$ 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;

k:= MEERCONS[0]:= read;

for j:= 1 step 1 until k do MEERCONS[j]:= read; W[0]:= 0;

for k:= 1 step 1 until 32 do

for j:= 1 step 1 until 32 do comprimatie[k,j]:= - j;

for k:= 1 step 1 until 32 do comprimatie[k,0]:= - 100;

for k:= 1 step 1 until 26 do comprimatie[27,k]:= k;

```

comprimatie[1,21]:= comprimatie[1,9]:= comprimatie[9,5]:=
comprimatie[5,5]:= comprimatie[5,9]:= comprimatie[15,9]:=
comprimatie[15,15]:= comprimatie[15,21]:= comprimatie[21,9]:= 28;
comprimatie[3,8]:= 29; comprimatie[14,7]:= 31
end;

```

```

procedure splits(n); value n; integer n;
begin integer a, e, i, o, u, j, k, l, s, t, y, ng, x, letter,
    volgende letter, eerste klinker, tweede klinker, woordbegin,
    wordeind, koppelteller, av koppelteller, aantal cons,
    meercons, langste;
    real av, vv;
    integer array VERWIJZING, w[1:50], av koppel[1:5];
    array SLOT, VOOR[2:5];

```

```

procedure splits af(letternr); value letternr; integer letternr;
begin koppelteller:= koppelteller + 1;
    koppel2[koppelteller]:= VERWIJZING[letternr];
    woordbegin:= letternr + 1; goto RESTWOORD
end;

```

```

boolean procedure klinkers;
begin klinkers:= false; eerste klinker:= tweede klinker;
    for j:= eerste klinker + 1 step 1 until wordeind do
        begin letter:= w[j];
            if letter = 28  $\vee$  letter = a  $\vee$  letter = e  $\vee$  letter = i  $\vee$ 
            letter = o  $\vee$  letter = u  $\vee$  letter = y then
                begin klinkers:= true; tweede klinker:= j; goto end end
            end;
    end;
end:
end;

```

```

INIT: l:= j:= koppelteller:= av koppelteller:= 0; letter:= W[1];
      woordbegin:= a:= 1; e:= 5; i:= 9; o:= 15; s:= 19; t:= 20; u:= 21;
      x:= 24; y:= 25; ng:= 31;
COMPR: j:= j + 1;
COMPR1: l:= l + 1;
        volgende letter:= comprimatie[letter,W[l + 1]];
        if volgende letter > 0 then
          begin letter:= volgende letter; goto COMPR1 end else
            begin w[j]:= letter; VERWIJZING[j]:= 1;
              letter:= - volgende letter; if l < n then goto COMPR
            end;
        wordeind:= j;
ACHTERVGSL: tweede klinker:= 0; klinkers;
          if ¬klinkers then goto AFWERKING;
          langste:= if wordeind > 5 then 5 else wordeind;
          av:= w[wordeind];
          for k:= 2 step 1 until langste do SLOT[k]:= av:= av × 50 +
            w[wordeind - k + 1];
          for k:= langste step - 1 until 2 do
            begin av:= SLOT[k];
              for j:= AV[k,0] step - 1 until 1 do if av = AV[k,j] then
                begin wordeind:= wordeind - k;
                  if wordeind < 1 then goto AFWERKING;
                  av koppelteller:= av koppelteller + 1;
                  av koppel[av koppelteller]:= VERWIJZING[wordeind];
                  goto ACHTERVGSL
                end
              end;
            goto RESTWOORD1;
RESTWOORD: if ¬klinkers then goto AFWERKING;
RESTWOORD1: aantal cons:= tweede klinker - eerste klinker - 1;
          if aantal cons = 0 then splits af(eerste klinker);
VOORVGSL: if wordeind > 4 then
          begin langste:= if tweedeklinker - woordbegin > 5 then 5 else
            tweedeklinker - woordbegin; vv:= w[wordbegin];

```

```

for j:= 2 step 1 until langste do VOOR[j]:= vv:= vv × 50 +
w[woordbegin + j - 1];
for k:= langste step - 1 until 2 do
begin vv:= VOOR[k];
  for j:= VV[k,0] step - 1 until 1 do if vv = VV[k,j] then
    splits af(woordbegin + k - 1)
  end
end;
if w[tweedeklinker - 1] = s ∧ ((w[tweedeklinker] = u ∧
w[tweedeklinker + 1] = t) ∨ (w[tweedeklinker] = i ∧
w[tweedeklinker + 1] = 14)) then splitsaf(tweedeklinker - 1);
if w[eersteklinker + 1] = ng then
begin if eersteklinker = 1 then
  begin VERWIJZING[2]:= 2; splitsaf(2) end;
  if aantal cons = 1 then splitsaf(eersteklinker + 1);
  if w[eersteklinker] = i ∧ w[eersteklinker + 2] = s then
    splitsaf(eersteklinker + 2)
  end;
if w[eersteklinker + 1] = x then splitsaf(eersteklinker + 1);
if eersteklinker > 2 ∧ aantal cons > 1 then
begin if w[eersteklinker - 1] = i ∧ w[eersteklinker] = o ∧
  w[eersteklinker + 1] = 14 ∧ w[eersteklinker + 2] = s then
    splitsaf(eersteklinker + 2)
  end;
if 2 ≤ aantal cons then
begin meercons:= W[tweede klinker - 1] × 50 + w[tweede klinker-2];
  for j:= MEERCONS[0] step - 1 until 1 do if meercons =
    MEERCONS[j] then splits af(tweede klinker - (if j < 4 ∧
    w[tweedeklinker - 3] = s then 4 else 3));
  if meercons = 616 ∧ w[tweedeklinker - 3] = s then
    splitsaf(tweedeklinker - 3);
  if meercons = 1019 ∧ aantal cons > 2 then
    splitsaf(tweedeklinker - 3);
  end;
splits af(tweede klinker - 2);

```

```

AFWERKING: for k:= av koppelteller step - 1 until 1 do
  begin koppelteller:= koppelteller + 1;
    koppel2[koppelteller]:= av koppel[k]
  end;
  koppel2[0]:= koppelteller; koppel2[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 = koppel2[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	-1	-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	-2	-2
29	-1	-3	-1	-1	-1	27	28		

achtervoegsels

5 achtervoegsels van twee letters

61	68	261	70	359
----	----	-----	----	-----

13 achtervoegsels van drie letters

50068	50062	10068	47564	33269
-------	-------	-------	-------	-------

45061	45068	27762	3519	3069
-------	-------	-------	------	------

17554	50258	50260		
-------	-------	-------	--	--

3 achtervoegsels van vier letters

535062	878119	503519		
--------	--------	--------	--	--

4 achtervoegsels van vijf letters

26803407	127253111	44925551	83171019	
----------	-----------	----------	----------	--

voorvoegsels

5 voorvoegsels van twee letters

105	1070	464	269	209
-----	------	-----	-----	-----

18 voorvoegsels van drie letters

48754	15768	30057	38470	30081
-------	-------	-------	-------	-------

40268	15762	10469	47563	63520
-------	-------	-------	-------	-------

10068	6061	48254	47561	51069
-------	------	-------	-------	-------

10457	53316	40915		
-------	-------	-------	--	--

13 voorvoegsels van vier letters

2763411	788420	788413	502869	263519
---------	--------	--------	--------	--------

788111	2378520	2397806	2752857	1502869
--------	---------	---------	---------	---------

1502857	2545769	2502869		
---------	---------	---------	--	--

5 voorvoegsels van vijf letters

13170369	39762719	121503419	81410961	120162869
----------	----------	-----------	----------	-----------

17 elementen van tweecons

916	920	911		
-----	-----	-----	--	--

602	902	904	169	420
-----	-----	-----	-----	-----

606	906	1119	619	507
-----	-----	------	-----	-----

569	1469	519	520	
-----	------	-----	-----	--

§4. Resultaat

Het programma uit §3 werd toegepast op 400 woorden uit vier Zweedse kranten [4] die aan het eind van de regel waren gesplitst. Er werd gecontroleerd of deze door de zetter aangebrachte splitsingen ook door het programma werden voortgebracht.

Dit bleek voor 14 woorden, ofwel $3\frac{1}{2}\%$, niet het geval te zijn. Hierbij dient opgemerkt dat in enkele van deze gevallen de krantensplitsing van dubieuze kwaliteit was.

Het eigenlijke foutenpercentage ligt waarschijnlijk lager omdat de zetter 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 Zweedse tekst.

- [1] H. Brandt Corstius, Automatisch tellen en scheiden van Nederlandse lettergrepen, Mathematisch Centrum, MR 67, 1964.
- [2] H. Brandt Corstius en E.G.M. Broerse, Automatisch scheiden van Duitse lettergrepen, Mathematisch Centrum, NR 2, 1967.
- [3] Skrivregler, Skrifter utgivna av Nämnden för svensk språkvård, 5e druk, 1965.
- [4] Dagens Nyheter, Göteborgs handels- och sjöfarts-tidning, Sydsvenska Dagbladet Snällposten, Svenska Dagbladet. Alle van 5 april 1968.

Summary

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

The program fails for $3\frac{1}{2}\%$ of the words in Swedish newspapers that were actually hyphenated by the typesetter.

The error percentage is acceptable for application in automatic typesetting of Swedish text.