fulde spørgere, der lige så ofte som vi andre ser disse ordforbindelser skrevet sammen, bliver spist af med at det er en slags fejl, men at den dog er så mikroskopisk at nævnet næsten ikke gider tale om den. Og på dette område er der kun foretaget ganske få ændringer i RO86; bl.a. skal man nu generelt særskrive præpositionsudtrykkene når der er en styrelse, også i "uegentlig" betydning. Men alligevel! Hvad med lidt valgfrihed her! Det ville dog passe bedre med nævnets almindelige linje at tage hensyn til en udbredt praksis blandt i øvrigt omhyggelige og rutinerede sprogbrugere.

Hvorfor nævnet er underlig stædigt på dette punkt har Allan Karker gjort udførligt rede for i Nyt fra Sprognævnet, nr. 3 1986. Konsekvenserne ville blive uoverskuelige hvis der blev lukket op for godteposen. Karker skriver: "Til eksempel måtte sammenskrivningen igår for en systematisk betragtning medføre ikke blot idag, imorgen, iforgårs, iaftes, men også iforgårsaftes, iovermorgeneftermiddag osv." Okay, okay, det er godt nok et uhyggeligt perspektiv. Men lad os få en eneste lille valgfrihed endnu: Lad os få lov til at skrive udover, overfor, indenfor osv. som rigtige præpositioner i ét ord! Og lad så bare dem der kan kende en styrelse når de ser den, fortsætte med særskrivningen.

Mette Kunøe og Erik Vive larsen (udg.):
2. Møde om Udforskningen af Dansk Sprog
Århus 1989

TONIVEAUBESKRIVELSE AF DANSK MORFOLOGI

Præsentation af en persondatamatversion af Kimmo Koskenniemis datamorfologiske model, anvendt til beskrivelse af nudansk skriftsprogsmorfologi.

Af Søren Brandt

1. Datalingvistiske modeller

Når man diskuterer forskellige grammatikker og sprogteorier (sprogmodeller), der beskriver det samme sproglige materiale, kan man klassificere dem i forhold hvor 'adækvate' de er, hvilket groft sagt angiver, i hvor høj grad de udtrykker generelle regelmæssigheder. Dette er ikke stedet til en nøjere gennemgang af de pågældende begrebsdannelser, og vi skal indskrænke os til at postulere én mulig karakterisering:

1. En iagttagelsesadækvat grammatik opregner de acceptable sproglige størrelser og tilskriver dem en konstituentanalyse. De pågældende størrelser vil være sætninger i den syntaktiske del af grammatikken og ordformer i den morfologiske del.

2. En beskrivelsesadækvat grammatik er iagttagelsesadækvat og tilskriver endvidere de sproglige størrelser en (dybde)strukturel analyse.

3. En forklaringsadækvat grammatik er beskrivelsesadækvat og baserer endvidere de strukturelle analyse på almensproglige (universelle) lovmæssigheder.

Det er klart, at man ud fra et (almen)lingvistisk synspunkt vil stræbe efter at udvikle grammatikker der er adækvate på et så højt niveau som muligt. Man skulle derfor tro, at hvis man som grammatiksprog, metasprog, benytter en datamatisk specifikation i stedet for en mere traditionel specifikation med et naturligt sprog som metasprog, så ville også en sådan datalingvistisk model være bedre, jo højere dens adækvansniveau var. Det er imidlertid ikke nødvendigvis tilfældet. En forklaringsadækvat datalingvistisk model må nemlig være baseret på en konkret, forklaringsadækvat sprogmodel, og den vil derfor være bundet til netop denne sprogmodels begrebsdannelser og forudsætninger, og det vil følgelig være umuligt at benytte den pågældende datalingvistiske model i forbindelse med nogen anden sprogmodel.

Nu véd vi jo forbavsende lidt om de almensproglige regelmæssigheder som vores grammatikker ideelt set skulle afspejle, og den lingvistiske debat handler især om de forskellige almensproglige teorier og deres respektive fortrin og mangler. Hvis vi derfor betragter en datalingvistisk model som et stykke værktøj, så er det klart, at det vil være mere velegnet, jo flere forskellige almensproglige modeller det kan benyttes i forbindelse med. Hvis man vil benytte en datalingvistisk model til at afprøve almensproglige modeller, vil man foretrække en model der er mest muligt teorineutral og som

følgelig uden større vanskeligheder kan benyttes i forbindelse med forskellige teoridannelser og forskellige varianter af en bestemt teoridannelse.

Det betyder at den datalingvistiske model skal være mindst mulig restriktiv. Mens man i en almensproglig model søger de mest mulig restriktive formuleringer for at opnå en model der er så karrig som mulig, d.v.s. så falsificérbar som mulig, vil man i en datalingvistisk model søge de mindst mulig restriktive specifikationsmetoder for at opnå en model der er så frodig som mulig, således at hver af de almensproglige modeller der vil anvende den, kan vælge sit eget (karrige) udsnit af de beskrivelsesmuligheder modellen stiller til rådighed. Man må altså ikke forveksle en datalingvistisk model med en sprogmodel: der kan godt, som her, argumenteres for at der bør stilles modsatrettede krav til de to modeltyper.

På den anden side skal en datalingvistisk model naturligvis helst være iagttagelsesadækvat; ellers kan man ikke rigtig bruge den til noget. I hvert fald skal dens eventuelle begrænsninger være veldefinerede og indgå i beskrivelsen af den pågældende model. De begrænsninger den datalingvistiske model besidder, vil samtidig være begrænsninger på, hvilke almensproglige teorier den kan benyttes i forbindelse med, og disse begrænsninger afspejler altså i virkeligheden postulerede eller forudsatte almensproglige regelmæssigheder.

Den datalingvistiske model vi her skal diskutere er mere specifikt en datamorfologisk model, d.v.s. en model hvori man kan beskrive forskellige sprogs morfologi. Den såkaldte toniveaumorfologi (Two-Level Morphology, TLM) er udviklet af den finske datalog Kimmo Koskenniemi, og det er næppe helt tilfældigt at han netop er finne: finsk har ca. 12.000 bøjningsformer af verberne og ca. 2.000 bøjningsformer af substantiverne. Denne model har i praksis vist sig i hvert fald iagttagelsesadækvat for så forskellige sprog som finsk, japansk, rumænsk, fransk og engelsk (Koskenniemi 1983a:137) samt svensk og oldkirkeslavisk (Koskenniemi 1983b:154). Hertil kan vi føje dansk, idet jeg har implementeret to udvidede versioner af den på forskellige mikrodatamater (Brandt 1985, Brandt 1988). Denne model kalder jeg XTLM (Extended Two-Level Morphology), i (Brandt 1988) dog af tekniske grunde MXTLM. Forskellen mellem TLM og XTLM ligger hovedsagelig i, at XTLM benytter en mere lettilgængelig brugergrænseflade, så det er lettere at læse - og skrive - de morfologiske specifikationer. Der er også større metasyntaktiske variationsmuligheder; men derimod er der ikke tilføjet væsentligere teoretiske udvidelser, og en XTLM-specifikation vil altid kunne omskrives til en - typisk noget mere kompliceret - TLM-specifikation.

2. Kort introduktion til toniveaumorfologi

2.1 De to repræsentationsniveauer og forbindelsen mellem dem

Toniveaumodellen er baseret på et leksikonsystem og et kompleks af morfologiske regler som relaterer de leksikalske repræsentationer til overfladeformer. TLM er karakteristisk derved, at reglerne i systemet er parallelle og ikke som konventionelle generative regler appliceres efter hinanden. Denne parallellitet, i sammenhæng med at der kun er netop to repræsentationsniveauer som kædes sammen, gør TLM til et egentligt tovejssystem, idet regelformuleringerne hverken behøver at læses som specifikationer af hvordan overfladeformerne deriveres fra de leksikalske former eller som specifikationer af hvordan de leksikalske former fremanalyseres fra overfladeformerne; men simpelthen udtrykker en retningsneutral relation mellem de to repræsentationsformer. Relationen kan være 1-til-1, 1-til-N eller N-til-1.

De to niveauer som toniveaumorfologien henter sit navn fra er et leksikalsk niveau og et overfladeniveau. I lighed med mange fonologiske beskrivelser forudsætter TLM således at den fonetiske eller grafiske fremtrædelsesform for et ord kan være relateret til en abstrakt leksikalsk repræsentation. I den nu klassiske generative fonologi postuleres frit underliggende segmenter af endog meget abstrakt karakter, og overfladeformerne udledes heraf ved en række af sekventielt ordnede regler, således at de tidligere regler afleder mellemliggende former fra de underliggende, hvorefter senere regler eventuelt afleder nye mellemliggende former o.s.fr., indtil den ønskede overfladeform fremkommer. Det kan vises at de postulerede regler i visse tilfælde skal benyttes i en bestemt rækkefølge for at få de rigtige resultater, og det der især adskiller TLM fra konventionel generativ fonologi er, at man her arbejder med uordnede regler uden mellemliggende former.

En toniveaubeskrivelse kan derfor ikke på en simpel måde afledes af en beskrivelse i form af generative genskrivningsregler; men man må for hvert leksikalsk segment undersøge hvilke overfladerealisationer det i henhold til genskrivningsreglerne vil kunne få, og i hvilke kontekster. Hvis et segment undergår ændring ved flere regler, og hvis disses interaktion er kompleks, er det langt fra nogen simpel opgave at omdanne en traditionel fonologisk beskrivelse til den datamatiske specifikation som en toniveaubeskrivelse er. Når toniveaumodellen ikke desto mindre har vist sig alment anvendelig og rimelig lettilgængelig, skyldes det formentlig især at netop inden for morfologien og specielt måske inden for grafo-morfologien er (komplekse) regelinteraktioner ualmindelige, således at sammenhængen mellem leksikalsk segment og overfladesegment i mange tilfælde kun beskrives ved en enkelt regel.

Korrespondancerne mellem leksikalske segmenter og overfladesegmenter angives ved notationer af typen

```
d   e   T   0
:   :   :   :
d   0   d   e
```

hvor det øverste tegn repræsenterer det leksikalske niveau og det nederste repræsenterer overfladeniveauet. Vi skal betegne sådanne korrespondancer som segmentpar, og de morfologiske regler i TLM gør det muligt at beskrive restriktionerne på rækkefølgen af sådanne segmentpar i en parallelstreng bestående af en leksikalsk repræsentation og en overfladerepræsentation der tegn for tegn er synkroniseret med hinanden, idet der dog som allerede vist kan forekomme tegn i den ene streng, som ikke har konkret repræsentation i den anden og dér i stedet noteres med et specielt tegn, her 0, som altså betyder 'ingenting'.

De morfologiske regler i et TLM-system skal altså specificere de tilladte sekvenser af segmentpar i en parallelstreng som repræsenterer en mulig ordform i det pågældende sprog. Det er vigtigt at bemærke at reglerne omhandler netop segmentpar, idet dette har den effekt at regelbetingelserne kan referere til begge repræsentationsniveauer. Regelsyntaksen i (Koskenniemi 1983a) kan illustreres ved nedenstående regelformuleringer for reglen der beskriver alternationen mellem "t" og "d". "?" betyder 'hvad som helst'.

```
T       ?     ?                    T
:  <=>  :  __ :      (1a),         :   =>  __      (1b)
d       e     e                    t
```

I reglerne anføres det segmentpar reglen angår på venstre side, og i reglens højre side markerer symbolet __ det sted i parallelstrengen hvor det pågældende segmentpar forekommer, idet denne forekomst så nøjere kan specifi-

ceres ved en venstrekontekst og en højrekontekst, begge bestående af ingen, et eller flere segmentpar. Reglens to sider forbindes ved et regelsymbol som angiver en eller begge af de to regeltyper i systemet (Koskenniemi 1983a:36-39):

=> betegner at det pågældende segmentpar kan forekomme i den angivne kontekst. Sådanne regler kaldes her adgangsregler. For at et segmentpar skal kunne forekomme i parallelstrengen, skal mindst én adgangsregel for dette segmentpar være opfyldt.

<= betegner at det pågældende segmentpar skal forekomme i den angivne kontekst. Sådanne regler kaldes her bindingsregler. For at et segmentpar skal kunne forekomme i parallelstrengen, skal samtlige bindingsregler for dette segmentpar være opfyldt.

<=> betegner kombinationen af de to andre regeltyper og kaldes følgelig for kombinationsregler.

De foranstående regler kan herefter udlæses f.eks. som følger: Regel (1a) fastsætter at leksikalsk "T" kan og skal realiseres som "d" mellem to overflade-"e"-er (uanset disses leksikalske herkomst), mens regel (1b) tillader leksikalsk "T" realiseret som "t" overalt. At der er en undtagelse, fremgår af (1a), men ikke af (1b).

I det oprindelige TLM-system findes en række yderligere regelnotationskonventioner som først og fremmest tjener til at gøre det muligt at sammenskrive flere elementære regler til én. Der er bl.a. mulighed for at definere delalfabeter ('subsets'), og regelnotationen giver endvidere mulighed for at angive fakultative elementer, elementer der kan gentages én eller flere gange, o.lign. Vi skal ikke gå nærmere ind herpå, da vi i XTLM benytter en anden måde at nedskrive de morfologiske regler på.

2.2 Leksikonstrukturer

Vi har indtil nu omtalt relationerne mellem en leksikalsk repræsentation og en overfladerepræsentation uden at komme ind på hvor egentlig den leksikalske repræsentation kommer fra. Den stammer selvfølgelig fra et leksikon, eller rettere et leksikonsystem bestående af et antal større eller mindre enkeltleksikonner, kædet sammen i en træstruktur. Formelt er der ikke tale om en træstruktur, idet et enkeltleksikon udmærket direkte eller indirekte kan have sig selv som efterfølger; men det er mest frugtbart at anskue strukturen som et træ der så kan indeholde visse identiske delkomponenter.

Hvert enkeltleksikon indeholder en eller flere leksikalske enheder som hver igen indeholder (1) leksikalske oplysninger, (2) oplysninger om efterfølgerleksikonner og (3) evt. yderligere data. I XTLM-systemet er de leksikalske oplysninger delt i en segmentstreng og hvad vi betegner som en klassemarkørstreng, indeholdende grammatiske oplysninger. I det oprindelige TLM-system måtte denne type oplysninger indkodes mellem de segmentale data. Også den måde efterfølgerleksikonnerne angives på, er forskellig i de to implementeringer. I TLM henviser hver leksikalsk enhed til en fortsætter, som er en selvstændig liste over efterfølgerleksikonner. I XTLM kan der for hver leksikalsk enhed umiddelbart specificeres en liste over forbindere til efterfølgerleksikonner. Gennem sådanne sammenkædninger kan f.eks. komplekse bøjningsendelser opbygges af simplere dele, eller mulige afledninger og sammensætninger kan specificeres gennem den leksikalske struktur. XTLM-systemet benytter et særligt initialt leksikon og et særligt finalt leksikon til at signalere begyndelsen og slutningen af leksikonstrukturen.

Enkeltleksikonnerne kan i stedet for stammer eller bøjningsendelser indeholde vilkårlige orddele, hvilket man bl.a. vil kunne drage nytte af ved beskrivelsen af ord hvis stammer ændres under bøjningen som f.eks. "gymnasium, konto" o.lign. Man kan her opfatte de variable dele som stammedannende suffixer og opføre dem i særskilte alternationsleksikonner.

Før vi forlader leksikonbeskrivelsen skal vi diskutere karakteren af de segmenter de leksikalske strenge opbygges af. Man vil typisk benytte tre typer, svarende til forskellige enheder i den traditionelle fonologiske analyse (Koskenniemi 1983a:23-24):

1. Overfladesegmenter. De segmenter der kan optræde i overfladeudtryk, vil typisk også kunne forekomme i den leksikalske repræsentation og vil typisk manifesteres som 'sig selv'.

2. Arkifonemer og morfofonemer som benyttes til at beskrive morfologiske, morfofonologiske og fonologiske alternationer. De har to eller flere alternative overflademanifestationer, muligvis inklusive 0 (ingen overflademanifestation).

3. Morfologiske træk, også kaldet tekstmarkører, der dels benyttes som grænsesignaler og kan influere på visse morfologiske regler og dels kan indgå i bøjningsmønstre o.lign. og eksempelvis udløse aflyd eller omlyd i foranstående morfofonemer. De vil typisk ikke have nogen overflademanifestation.

Det er imidlertid ikke noget systemkrav at man skal bygge sine leksikalske strenge op af segmenter der ligner sædvanlige fonologiske størrelser. Der er f.eks. intet i vejen for at bruge vilkårlige symboler også for de segmenter der undtagelsesløst manifesteres på en bestemt måde; men det gør selvfølgelig leksikonspecifikationerne vanskelige at læse, herunder korrekturlæse, hvis man repræsenterer overflade-f ved et #-tegn i leksikonnet og til gengæld bruger leksikalsk f som ordgrænse.

De mulige segmenter defineres i XTLM ved hjælp af et alfabetprogram der indlæser en række specifikationer der hver angiver et segmentsymbol samt en eller flere eksterne repræsentationer for det. Man kan således, hvis man har en gammeldags mikrodatamat, indtaste sekvensen "ø:" og få den skrevet på printeren som "o,Baktegn,-", og man kan benytte flertegnsnavne til sine segmenter, f.eks. kan "I1, I2, I3" angive forskellige vokalalternationsmønstre. Der kan endvidere defineres navne på delmængder af alfabeter, herunder navne på 'hvad som helst' (f.eks. "?") og 'ingenting' (f.eks. "0"). Alfabetprogrammet omdanner specifikationerne til en række tabeller, der benyttes af de øvrige programmer i systemet.

De leksikalske specifikationer i XTLM behandles af en gruppe leksikonprogrammer der opbygger et internt leksikon i et kompakt format som det er rimeligt effektivt at afsøge.

2.3 Regelformalisme

Implementeringen af de morfologiske regler er baseret på, at det følger af TLM's regelformalisme at hver enkelt regel kan udtrykkes ved hjælp af en endelig tilstandsmaskine (finite state automaton), og konventionen om at reglerne anvendes parallelt gør det muligt at implementere hver regel som en selvstændig tilstandsmaskine, der uafhængigt af alle de øvrige følger parallelstrengen og kontrollerer at netop den tilstandsmaskines regelbetingelser er opfyldt. (I XTLM kontrolleres klassemarkørerne også, men det ser vi for nemheds skyld bort fra i den følgende fremstilling.)

Hele parallelstrengen bliver kun accepteret såfremt (1) ingen af maskinerne indtræder i en fejltilstand, d.v.s. med sikkerhed konstaterer at parallelstrengen ikke er acceptabel, og såfremt (2) samtlige tilstandsmaskiner er i en såkaldt sluttilstand når afslutningen på parallelstrengen nås, hvilket vil sige at de ikke har nogen manglende højrekontekst som betingelse for forekomsten af de segmentpar som strengen består af. En tilstandsmaskine kan defineres formelt som bestående af følgende elementer:

1. En endelig mængde af tilstande, som hver repræsenterer at en bestemt situation foreligger, i dette tilfælde at en bestemt sekvens af segmentpar indtil nu er blevet 'lagt mærke til' af den pågældende maskine.

2. Et endeligt alfabet af segmentpar, idet enhver kombination af et leksikalsk segment og et overfladesegment er et element i maskinens alfabet.

3. En tilstandsovergangsfunktion, der for hver kombination af tilstand og alfabetelement specificerer hvilken tilstand maskinen skal overgå til, hvis maskinen er i den pågældende tilstand, og det pågældende element er det næste segmentpar i strengen.

4. En starttilstand som maskinen befinder sig i, før det første segmentpar i strengen behandles.

5. En delmængde af tilstandsmængden: sluttilstandene, der repræsenterer de tilfælde hvor en parallelstreng (indtil videre) er blevet accepteret af den pågældende maskine. Tilstande der ikke er sluttilstande repræsenterer tilfælde hvor der enten kræves bestemte efterfølgende betingelser opfyldt (ventetilstande), eller hvor det allerede med sikkerhed er konstateret at parallelstrengen ikke kan accepteres (fejltilstande).

   Ventetilstande og fejltilstande adskiller sig fra hinanden derved at tilstandsovergangsfunktionen for en fejltilstand altid, uafhængig af de følgende segmentpar, fører over i samme tilstand igen. Derimod er der i en ventetilstand mindst ét segmentpar der fører over i en anden tilstand. Det følger heraf, at hvis blot én af tilstandsmaskinerne går ind i en fejltilstand er det unødvendigt at behandle parallelstrengen videre. I stedet for at benytte fejltilstande lader vi derfor tilstandsmaskinerne signalere fejl så snart en sådan situation opstår.

En tilstandsmaskine kan repræsenteres som en matrix hvor hver række repræsenterer en bestemt tilstand og hver søjle en bestemt type segmentpar (der checkes fra venstre mod højre), og hvor matrixelementerne udgør tilstandsovergangsfunktionen, d.v.s. indeholder nummeret på den nye tilstand. Benyttes 0 for at markere fejlsignalering kan en tilstandsmaskine for t/d-alternationen repræsenteres ved følgende matrix, hvor en '*' efter tilstandsnummeret angiver sluttilstande:

	? e	T d	T t	+ ?	? ?
1 *	2	0	1	1	1
2 *	1	3	5	1	1
3	0	0	0	4	0
4	1	0	0	0	0
5 *	2	0	1	6	1
6 *	0	0	1	1	1

I den oprindelige TLM-implementation indlæses de morfologiske regler faktisk - tro det om man vil - på denne form, idet regelformalismen omsættes med håndkraft til tilstandsmaskiner på matrixform i henhold til retningslinier som gives i (Koskenniemi 1983a:103-107), og denne metode har altså også vist sig gennemførlig i praksis, men som det anføres 'The need for a rule compiler is, however, obvious', og der er da også allerede samme sted i disputatsen skitseret en sådan, som senere er blevet udviklet (Koskenniemi 1985).

I XTLM-systemet benytter vi en helt anden metode, idet vi beskriver tilstandsmaskinerne i et symbolsk sprog. Hver tilstand angives ved et symbolsk navn, og for hver tilstand angives tilstandsovergangsfunktionen som en ordnet sekvens af vilkår (segmentpar eller klasser af segmentpar) med tilhørende overgangstilstand. Der er adskillige raffinementer hertil som vi imidlertid ikke skal gå ind på. Regelspecifikationerne oversættes til en intern repræsentation med et regelprogram.

3. Leksikalsk beskrivelse af talord

3.1. Forudsætninger

Der benyttes et alfabet bestående af nedennævnte enheder med tilhørende realisationsregler:

a - å	realiseres som sig selv
J	realiseres som "i" eller "j" efter frit valg
N	realiseres som "t" foran et endelses-T, ellers som "n"
=	realiseres identisk med den foregående konsonant, hvis denne forudgås af en vokal, og der også følger en vokal efter
+	indleder endelser; realiseres som ingenting.
#	adskiller ord; realiseres som ingenting
&	realiseres som mellemrum eller (normstridigt) ingenting

Der antages følgende morfologiske regler:

+e	bortfalder efter "e"
+t	bortfalder efter "t" (også når det stammer fra leksikalsk N)
%' og %S	realiseres efter reglerne for genitivendelser (R086:551)

3.2. Indlejringsfiler

NUM/EN.DIC

eN	NumCard	NumTE

NUM/ET.DIC

eN+t=	NumCard	

NUM/CARD.DIC

to	NumCard	
tre	NumCard	
fire	NumCard	
fem=	NumCard	NumTE
seks	NumCard	
syv	NumCard	NumENDE
ot=e	NumCard	NumENDE
ni	NumCard	NumENDE

NUM/ORD.DIC

```
O:   første          S
O:   andeN           PronT
     tredJe          NumOrd
     fjer            NumDE
     sjet            NumTE
```

3.3. Hovedleksikon

```
MAIN LEXICON   Talord:        NumOg
.FI  NUM/EN.DIC

MAIN LEXICON   Talord:                                  NumHund    NumTus
.FI  NUM/ET.DIC

MAIN LEXICON   Talord:        NumOg      NumTi          NumHund    NumTus
.FI  NUM/CARD.DIC

MAIN LEXICON   Talord
.FI  NUM/ORD.DIC

MAIN WORDLIST  Talord:                   NumTi
     fir, ot

MAIN LEXICON   Talord
     ti                       NumCard    NumENDE                   NumTus
     elv                                 NumTE
     ellev                               NumTE
     elleve                   NumCard                 NumHund    NumTus
     tolv                     NumCard    NumTE        NumHund    NumTus

MAIN WORDLIST  Talord:        NumCard    NumDE        NumHund    NumTus
     tretten, fjorten, femten, seksten, sytten, atten, nitten

MAIN LEXICON   Talord20
     tyve                     NumCard    NumENDE                   NumTus

MAIN LEXICON   Talord30-90
     tredv                               NumTE
     trediv                              NumTE
     tredive                  NumCard                              NumTus
     fyrre                    NumCard    NumTyve                   NumTus

MAIN WORDLIST  Talord30-90:              NumSind
     halv#tred, tre, halv#fjerd, fir, halv#fem

MAIN LEXICON   Talord100
     hundrede                 NumE

MAIN LEXICON   Talord1000
     tusind                   NumE
```

3.4. Morfologiske leksikonner

```
NUM  LEXICON        NumTyve
     #              Talord20
```



```
NUM  LEXICON        NumSind
     s=                       NumCard
     #sind+s#                 Talord20

NUM  LEXICON        NumOg
     #og#                     Talord20  Talord30-90

NUM  LEXICON        NumHund
     &                        Talord100

NUM  LEXICON        NumTus
     &                        Talord1000

NUM  LEXICON        NumTi
     #ti                      NumENDE
     #ti#                     NumNord

NUM  LEXICON        NumNord
.FI  NUM/EN.DIC
.FI  NUM/ET.DIC
.FI  NUM/CARD.DIC
.FI  NUM/ORD.DIC

NUM  LEXICON        NumTE
     +te                      NumOrd

NUM  LEXICON        NumE
C:   ""                       S
     +e                       NumOrd

NUM  LEXICON        NumDE
     +de                      NumOrd

NUM  LEXICON        NumENDE
     +ende                    NumOrd

C:   NUM  PATTERN   NumCard
                              S     NumER

NUM  LEXICON        NumER
     +er                      N/NE2

O:   NUM  PATTERN   NumOrd
                              S     NumDel

NUM  LEXICON        NumDel
     #del                     N/NE1
```

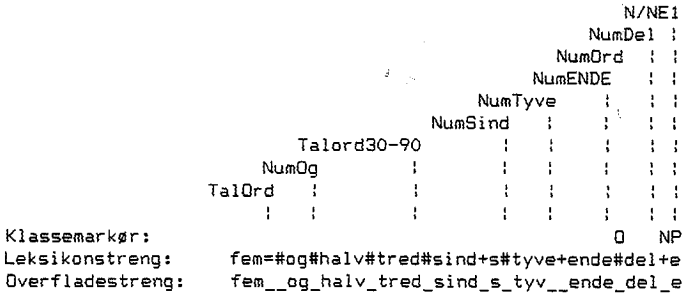
3.5. Bøjningsleksikonner

```
SFX  LEXICON        S:        #     ; # er navnet på det finale leksikon
     ""
G:   %'%S

SFX  LEXICON        PronT:    S
     ""
T:   +t
```

```
N:   SFX  LEXICON   N/NE1:    S
     ""
D:   +en
P:   +e
PD:  +ene

N:   SFX  LEXICON   N/NE2:    S
     ""
D:   +en
P:   +e
PD:  +ne
```

3.6 Eksempel

I overfladestrengen er 'ingenting' her markeret som '_' for at synkronisere den med leksikonstrengen. Klassemarkørstrengen er også vist synkroniseret med leksikonstrengen for overskuelighedens skyld, men er det ikke i praksis.




```
Klassemarkør:                                    0   NP
Leksikonstreng:     fem=#og#halv#tred#sind+s#tyve+ende#del+e
Overfladestreng:    fem__og_halv_tred_sind_s_tyv__ende_del_e
```

4. Alfabetdefinition og morfologiske regler

4.1 Alfabetdefinition (uddrag)

```
COMMON ALPHABET          ; Normal alphabetic characters

*    Upper case letters are used for alternating characters in the
*    primary alphabet, but stand for themselves in the secondary
*    alphabet.  Capitalization is signaled in the dictionary by the
*    flag characters !n.

UpCase:       A
     ...
UpCase:       Å

LoCase:       a
     ...
LoCase:       å
```



```
Spec:          '
Space: Spec:   -
Spec:          .
Spec:          "/"
Space:         Blk  " "        ; Required blank

PRIMARY ALPHABET               ; Special blank characters

Spec:          ('}             ; Optional quote
Space: Spec:   (-)             ; Optional hyphen
Space:         &               ; Prescribed blank, often omitted
Space:         ()              ; Optional blank
Space:         )(              ; Unprescribed blank often inserted

*    The genitive ending is "%'%S", realized as "s" after non-sibilants, and
*    as "'", "'s" or (rarely) "es" after sibilant letters (S, Z, X)

               %'
               %S

COMMON ALPHABET                          ;  Markers

Null:          0                         ; Null character

Store:         #1                        ; FSA memory unit
Store:         #2                        ; FSA memory unit

Nil:              /   _                  ; Only keyboard value, omitted unit

                                         ; Capitalization markers
Feat:          !1   /    !,  !1          ; InitCaps
Feat:          !2   /    !!, !2          ; AllCaps

PRIMARY ALPHABET

Feat:          #               ;  Word boundary
Feat:          +               ;  Morpheme boundary
Feat:          [               ;  Precedes silent letters at end of word
Feat:          =               ;  Doubling stem
Feat:          *               ;  Syncope stem
Feat:          $               ;  Represents final "." in abbreviations

COMMON SUBSET

Any:           ?                         ;  Any character whatever
UpCase:        Upp
LoCase:        Low
Spec:          Spc                       ;  Special characters in lexicon

               Vowl      a e i o u y æ ø å A E I O U Y Æ Ø Å

               Cons      b c d f g h j k l m n p q r s t v w x z
               Gem       b d f g k l m n p r s t          ; Geminating
               Sib       s S x X z Z                      ; Sibilants
```

```
PRIMARY SUBSET

Space:          SP
Feat:           ~
```

4.2 Korrespondancedefinitioner (uddrag)

```
    PAIRS

    Z           z    s    Z    S
    Low         Low
    a           A
      ...
    å           A

    (')         0    '
    (-)         0    -

;   The genitive ending consists of the two units "%'%S" and is realized as
;               ' or 'S         after silent sibilants
;               ES, ' or 'S     after pronounced sibilants
;               S               after non-sibilants

    %'          0    e    s
    %S          0    s

    END
```

4.3 Morfologisk regel for genitivformer

```
Genitive: RULE ; Realization of genitive endings

!Reset:   [    ?    Silent    ; Special rules if silent final letter
          ?    Sib  AudSib    ; Pronounced sibilant letter
          %'   0    ReqS      ; No ' allowed
          %'   ?    Error

!Silent:  ?    Sib  SilSib    ; Silent sibilant letter
          %'   0    ReqS      ; No ' allowed
          %'   ?    Error

!AudSib:  ~    0    Loop
          %'   e    ReqS      ; Endings ES,
          %'   '    OptS      ; or ' or 'S allowed
          %'   0    Error
          else       Begin     ; Not genitive

!SilSib:  ~    0    Loop
          %'   '    OptS      ; Endings ' or 'S allowed
          %'   ?    Error
          else       Begin     ; Not genitive
```



```
?ReqS:    ?    s    Reset     ; Secondary S must follow
          else      Error

!OptS:    else      Reset     ; Accept secondary S or 0

          END
```

5. Status og bestik

Vi har ovenfor illustreret nogle af de beskrivelsesteknikker der kan benyttes i XTLM, og vi skal nu prøve at vurdere toniveaumodellen og fremdrage nogle punkter hvor den vil kunne forbedres. Det er naturligvis ikke vanskeligt at opfinde yderligere beskrivelsestekniske finesser man kunne ønske sig; men vanskeligheden består i at begrænse sig til forbedringer, der ikke angriber de grundstrukturer i toniveaumorfologien, der har gjort det muligt at konstruere rimelig effektive datamatiske implementeringer af den. Af disse grundstrukturer skal navnlig fremhæves to: (1) Den leksikalske beskrivelse gør det muligt at opbygge særdeles kompakte leksikondatasæt, der ikke desto mindre kan afsøges med acceptabel hastighed. (2) Den morfologiske komponent er simpel nok til at kunne implementeres uden brug af avancerede programteknikker og kræver kun ubetydelige ressourcer i form af arbejdslager i datamaten.

TLM-modellen bygger principielt på den grundantagelse, at ordformer kan beskrives ved kontekstfri grammatikregler af typen

<Non-terminal> ::= <Præ-terminal> (<Non-terminal>)

hvor højresidens <Non-terminal> kun kan udelades i det finale leksikon, d.v.s. det leksikon der afslutter alle leksikonstrukturer. I alle andre enkeltleksikonner vil <Non-terminal> være den mængde af enkeltleksikonner, som forbinderne henviser til, og <Præ-terminal> vil være den mængde af leksikalske strenge (i XTLM incl. klassemarkører), der benytter netop denne forbindermængde. I den lingvistiske praksis er det imidlertid langt mere almindeligt også at benytte regler af formen

<Non-terminal> ::= <Præ-terminal> <Non-terminal> <Præ-terminal>

altså regler der tillader indlejring. Det er simpelt at påvise, at TLM-modellen ikke kan anvendes til ubegrænsede indlejringskonstruktioner, mens man derimod godt kan specificere indlejrende konstruktioner, såfremt man kan begrænse sig til et forudbestemt antal indlejringsniveauer. Det er også klart, at regler af typen

<Non-terminal> ::= <Non-terminal> <Præ-terminal>

der forbinder 'baglæns', ikke umiddelbart kan nedskrives i et TLM-leksikon. Man skal derimod fremfinde de enkelte præfigerede (nonterminale) enkeltleksikonner og give dem forbindere til det præfigerende (præterminale) leksikon reglen omhandler.

De nævnte to vanskeligheder kan begge afhjælpes ved at indføre en yderligere forbehandlerfunktion der tillader regler af de viste typer (indlejrende, hhv. præfigerende) og automatisk omsætter dem til de korrekte forbinderspecifikationer. Hvis samtidig vi erstatter regelsystemets tilstandsautomater med stakautomater, en ændring der ikke er vanskelig at gennemføre, vil systemet i princippet også kunne håndtere ubegrænset indlejrende konstruk-

tioner. Disse ændringer vil ikke have nævneværdig indflydelse på systemets ressourcekrav og vil forenkle dets brug mærkbart.

Vi har dog i beskrivelsen af dansk morfologi indtil nu (Brandt 1988) ikke fundet nogen fænomener der ikke med større eller mindre indsats kan beskrives i XTLM, og de nævnte raffinementer er derfor næppe af påtrængende karakter.

Litteraturreferencer

Brandt 1985
Brandt, Søren: Implementation of Two-Level Morphology on a Microcomputer. Nordic Linguistic Bulletin 9, nr. 2, s. 6 - 9.

Brandt 1988
--- : En model til automatisk morfologisk analyse. Lambda 8, Institut for Datalingvistik, Handelshøjskolen i København.

Koskenniemi 1983a
Koskenniemi, Kimmo: Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production. Publication no. 11, Department of Linguistics, University of Helsinki.

Koskenniemi 1983b
--- : A General Computational Model for Word-Form Recognition and Production. Sågvall Hein, Anna (ed.): Föredrag vid de nordiska dataling-vistikdagarna 1983, Centrum för datorligvistik, Uppsala Universitet, 1984, pp. 145 - 154.

Koskenniemi 1985
--- : Compilation of Automata from Morphological Two-Level Rules. Karlsson, Fred (ed.): Papers from the Fifth Scandinavian Conference of Computational Linguistics. Publication no. 15, Department of Linguistics, University of Helsinki, 1986, pp. 143 - 149.

RO86
Dansk Sprognævn: Retskrivningsordbogen. Gyldendal.

Mette Kunøe og Erik Vivé Larsen (udg.):
2. Møde om Udforskningen af Dansk Sprog
Århus 1989

ER DET DANSKE SPROGS MORFOLOGI 'NATURLIG'?

Af Kurt Braunmüller

1. Teorien om den 'morfologiske naturlighed'

Teorien om den morfologiske naturlighed (MN), også kaldet teorien om den naturlige morfologi, har sit udgangspunkt i Stampes arbejder om fonologisk naturlighed (jf. Stampe 1969).

Mens det er nogenlunde uproblematisk at begrunde begrebet 'naturlig' inden for rammerne af en fonetisk eller en fonologisk teori, så har anvendelsen af dette begreb på morfologien derimod hos mange ført til misforståelser og forkerte associationer. Begrebet 'naturlig' skal i denne sammenhæng ikke forstås som 'konkret, enkel, intuitiv plausibel' eller som modsætning til 'kunstig'. Derimod står begrebet 'naturlig' som fagterminus brugt om en meget kompleks teori om morfologi; en teori, som her kort skal karakteriseres.

Teorien om MN inddrager data fra mindst fire forskellige områder: (a) den almene semiotik, (b) sprogtypologi og forskning i sproglige universalier, (c) undersøgelser af sprogforandring samt (d) data angående sprogtilegnelse (hos børn). Teorien om den MN gør således krav på at give en omfattende forklaring af et bestemt enkeltsprogs morfologiske strukturering såvel som af den morfologiske strukturering af sprog overhovedet. De mest overbevisende resultater foreligger på de svagt flekterende sprogs område og specielt på de agglutinerende sprogs område. Større problemer er der stadig med