Bente Maegaard,

Københavns Universitet,
Institut for anvendt og
matematisk lingvistik,
Njalsgade 96
2300 København S

Regelformalismer til brug ved datamatisk lingvistik.

Når man i 'gamle dage' lavede et system til sproglig analyse, gjorde man det oftest på den måde, at man skrev et program, i hvilket man udtrykte al den viden, der skulle bruges. Det, der især er interessant her, er at den grammatiske viden der skulle bruges, var udtrykt i selve programmet. Programmøren og sprogforskeren var måske en og samme person, men selv i det tilfælde er det uhensigtsmæssigt - af mange kendte grunde.

Derfor er man da også mere og mere gået over til at skrive systemer, hvor program og data holdes adskilt - og grammatikker altså opfattes som data. Disse grammatikker skrives i et bestemt format; det er det jeg ovenfor har kaldt regelformalisme.

Der findes efterhånden en række sådanne parsere med forskellige regelformalismer. Disse formalismer afviger fra hinanden på forskellige måder. Jeg vil her især interessere mig for, hvordan det ser ud fra brugerens - lingvistens - synspunkt, og mindre for, hvordan det er implementeret.

De oplagte krav, som en lingvist stiller til en regelformalisme, er at den er rimeligt naturlig - man skal kunne udtrykke sproglige fakta på en rimeligt intuitiv måde - og at den er klar og overskuelig.

EUROTRA.

Det projekt, som jeg i det følgende vil referere til, er EUROTRA, EF's maskinoversættelsesprojekt. Projektet blev vedtaget i november 1982 og har en løbetid på 5,5 år. Efter 5,5 år skal en prototype af systemet være færdig; den skal kunne oversætte mellem de 7 EF-sprog (dansk, engelsk, fransk, græsk, italiensk, nederlandsk og tysk). De tekster, der skal

Regelformalismer til brug ved datamatisk lingvistik
Bente Maegaard
Proceedings of NODALIDA 1983, pages 162-168

kunne oversættes skal ligge inden for et bestemt emneområde (f.eks. informationsteknologi), med et ordforråd på ca. 20.000 ord.

Projektperioden er inddelt i 3 faser, á 2, 2 og 1.5 år. I den første fase, som vi nu er i gang med, skal både den lingvistiske og den programmelmæssige side - og koblingen mellem dem - defineres.

EUROTRAs regelformalisme.

Regelformalismen er en meget væsentlig del af grænsefladen mellem det sproglige og det programmelmæssige: man skal kunne udtrykke de sproglige fakta, som den lingvistiske model lægger op til, og samtidig bestemmer den valgte type af programmel en række egenskaber ved formalismen.

En hovedfilosofi i EUROTRA's programmelsystem er, at det skal være deklarativt. Hermed menes, at den lingvistiske viden om, hvad der er et sprogligt faktum, er adskilt fra den procedurelle viden om hvordan og hvornår denne viden skal udnyttes. I den yderste konsekvens betyder dette, at lingvisten skriver sine regler og at han ikke ved, i hvilken rækkefølge, de bliver anvendt. Det er dog næppe muligt at forestille sig f.eks. hele analysen af dansk skrevet i et stort ustruktureret deklarativt system. Derfor er EUROTRA's programmel bygget som et såkaldt 'Controlled Production System', dvs. et produktionssystem udstyret med et kontrolsprog. Kontrolsproget giver mulighed for at samle regler i 'grammatikker' og endvidere for at bestemme, om en grammatik skal anvendes kun én gang eller gentages, om grammatikker på samme niveau skal anvendes parallelt eller sekventielt osv. Dette er den mest procedurale del af formalismen.

Der gælder to hovedprincipper for den deklarative del af EUROTRA's regelformalisme: den skal være generel og den skal kunne beskrive de relevante data.

Kravet om generalitet skal forstås således: EUROTRA har 3 hovedmoduler: analyse, overførsel og generering, og det er hensigten, at samme formalisme skal kunne bruges i alle moduler. Denne formalisme skal yderligere kunne anvendes til

at udtrykke forskellige lingvistiske strategier, idet de deltagende grupper fra de forskellige lande nogenlunde frit skal kunne vælge strategi (inden for de rammer, som det valgte produktionssystem sætter).

Udover at formalismen skal kunne beskrive forskellige typer af lingvistisk strategi, skal den som nævnt kunne bruges på alle niveauer af oversættelsesprocessen: til grammatikregler såvel som ordbogsregler, til morfologi såvel som kasusgrammatik og syntaks, til regler med stor kompleksitet såvel som til helt enkle regler.

Dette hovedkrav om, at formalismen skal være generel er af to grunde delvis modstridende med kravet om naturlighed og klarhed. For det første må man tage hensyn til de mest komplekse regler, når man udformer formalismen, og det betyder, at de enkle regler kan blive unødigt komplicerede at udtrykke. For det andet er det jo sådan, at jo mere skræddersyet en formalisme er til et bestemt formål, jo nemmere er det at bruge den. Der er således gode argumenter for at udvikle særlige formalismer (og særlige fortolkere) til specielle velafgrænsede delopgaver. Man vil dog udarbejde en generel formalisme, der kan bruges overalt, således at eventuelle specialformalismer kun er et supplement.

Jeg har ovenfor nævnt to hovedkrav til formalismen:

1) den skal være deklarativ - og jeg har nævnt, at det ikke helt er opfyldt, og at det næppe heller er nogen god idé at hævde kravet rigoristisk,
2) den skal være generel. Dette krav kan opfyldes, men det er formentlig heller ikke her hensigtsmæssigt at overholde kravet strengt.

Det tredie hovedkrav er, at formalismen skal kunne håndtere de data, vi arbejder med i EUROTRA. Dette krav er der ingen mulighed for at slække på.

De data, der skal behandles, er træstrukturer med komplekse oplysninger (dekorationer) på kunderne. Træstrukturerne skal kunne se ud på alle mulige måder; men dekorationerne kan kun indeholde bestemte oplysninger i bestemte mønstre. Man kan derfor lade brugeren erklære, hvilke dekorationer, der

er mulige. Det er praktisk både for det fortolkende program og for brugeren.

Det generelle regelformat.

Det generelle format for en regel i EUROTRA-formalismen er den velkendte genskrivningsregel:
venstre side → højre side
Her består såvel venstre side som højre side af træstrukturer med knudeoplysninger. Formatet ser således ud:

geometry	< specifikation af et træ > ➡ < specifikation af et træ >
conditions	< betingelser på dekorationerne >
assignments	< tilskrivning af værdier til højresiden>

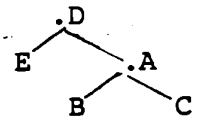
F.eks. vil

geometry	A + B ➡ C(A' + B')
conditions	MS of A = ADJ and
	MS of B = NOUN and
	GENDER of A = GENDER of B and
	NUMBER of A = NUMBER of B
assignments	A':=A;
	B':=B;
	MS of C:=NP
	GENDER of C:=GENDER of B;
	NUMBER of C:=NUMBER of B.

danne et substantivsyntagme af et adjektiv og et substantiv, der stemmer overens i køn og tal. (MS betyder morpho-syntactic class, resten skulle være umiddelbart forståeligt).
Det kan måske føles en lille smule omstændeligt at skrive regler på denne måde; men det kan næppe gøres meget nemmere.

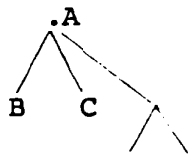
En af de ting, vi har diskuteret, er om træstrukturering i geometrien altid skal være et træ med rod eller om det godt kan være et deltræ af et større træ, altså om træet

også skal kunne findes i denne datastruktur



Sommetider vil lingvisten gerne have at træet bliver fundet, uanset hvor det er placeret, sommetider er han kun interesseret, hvis det har en rod. Det skal derfor være muligt at specificere dette. Her adskiller systemet sig fra f.eks. Q-systemet, hvor man kun kan få adgang til kunder i et træ ved at specificere hele vejen fra roden. Det at specificere hele denne vej er besværligt, men meget værre er det, at man må lave lige så mange regler, som der findes mulige træstrukturer, som deltræet kan indgå i.

Et andet særligt tilfælde, er noget svarende til følgende



hvor B og C ikke er de eneste datterknuder til A. Det er relativt nemt at klare at beskrive i en formalisme: i EUROTRA skriver man A(B+C +#X), hvor #X kan være tom eller bestå af et eller flere træer, hvis der kan være grene til højre for C. Hvis der også kan være grene til venstre for B og mellem B og C, må man skrive A(#Y+B+#Z+C+#X).

En sidste ting, jeg vil nævne, omkring træstrukturerne, er, at de selvfølgelig normalt er ordnede, dvs. i træet A(B+C) står B til venstre for C. Orden er som regel en relevant egenskab ved et træ. Men netop i oversættelsesprocessen er der ét tilfælde, hvor man evt. kan være uinteresseret i orden. Det er i genereringsfasen. Her har man ved udgangen

fra overførselsfasen fået en træstruktur, hvor ordene står i en eller anden rækkefølge, som ikke nødvendigvis er den rigtige på målsproget. Her vil det kunne være praktisk, at man kan skrive konstituenterne op, meddele at de må betragtes som uordnede, og fortælle, hvilken orden man ønsker, de skal stå i. Alternativet er, at man må lave lige så mange regler, som der er 'forkerte' rækkefølger for ordene; dette er for det første besværligt, for det andet betyder det, at man skal forestille sig alle de mulige rækkefølger af konstituenter, hvilket ofte vil føles helt irrelevant.

Jeg har her nævnt nogle af de muligheder, vi mener der er for at lette lingvistens arbejde med at skrive regler vedrørende træstrukturer. Selv om implementeringen af dem gør systemet lidt mindre effektivt, er der tale om en god investering.

Brugergrænsefladen.
Brugergrænsefladen består dels af den/de regelformalismer, systemet tilbyder og dels af de editeringsfaciliteter, der stilles til rådighed.

Brugerens arbejde kan lettes meget ved at specielle editorer stilles til rådighed. F.eks kan en regeleditor automatisk bede om de 3 hovedelementer i en regel, og en ordbogseditor kan automatisk bede om at få udfyldt relevante felter (afhængig af allerede indtastet information, således f.eks. at man beder om køn for substantiver, men ikke for verber). Sådanne editorer vil blive udarbejdet.

Når spørgsmålet om brugergrænseflade føles så vigtigt i dette projekt, er det fordi 100-150 mennesker fordelt på 10 lande og endnu flere universiteter, skal arbejde intensivt med den, når projektet er i gang i fuld skala.

Alt hvad der kan gøres for at lette lingvisternes arbejde skal gøres, for det første fordi det hurtigt vil have tjent sig ind, for det andet - og ikke mindst - fordi det giver større sikkerhed mod fejl.

Litteratur.
Alain Colmerauer: Les systèmes-Q, TAUM, Montréal, 1970.
Anna Sågvall Hein: A parser for Swedish, UCDL, Uppsala, 1983.
Dieter Maas and Bente Maegaard: Syntax and Semantics of the EUROTRA Formalism, EEC, 1984 (ikke frit tilgængelig).