

ATT KNYTA NORDENS SPRÅK TILL ETT MÅNGSPRÅKIGT DATORÖVERSÄTTNINGSSYSTEM

Klaus Schubert

BSO/Research
Postbus 8348
NL-3503 RH Utrecht
Nederländerna

Uppsala universitet, FUMS
Box 1834
S-751 48 Uppsala
Sverige

Elektronisk adress: schubert@dlt1.uucp

1. Ett mångspråkigt datoröversättningssystem

Jag beskriver här i korthet datoröversättningssystemet *Distributed Language Translation (DLT)* och tar i samband med detta upp Nordens språk. DLT är ett omfattande forsknings- och utvecklingsprojekt som bedrivs av det nederländska softwareföretaget Buro voor Systeemontwikkeling (BSO/Research i Utrecht) med anslag från Nederländernas Ekonomidepartement. Projektet är inne på en än så länge icke-kommersiell sjuårsperiod (1985-1991) som skall leda till en prototyp för ett översättningssystem för icke-litterär engelska och franska. Prototypen omfattar bara två språk, men DLT är från början beräknat att bli **mångspråkigt**, vilket innebär att det måste vara **modulärt utbyggbart**. Därför utförs redan nu i samarbete med forskare vid universitet i vederbörande länder och med andra experter förberedande studier om tillämpligheten av DLT:s grammatikmodell på andra språk (och även implementeringar i begränsat omfång). Bland dessa är även några av Nordens språk.

Det är möjligt att utvecklingen av utgångs- eller målspråkssystem för fler språk påbörjas inom DLT före 1991. Med detta perspektiv upptar jag här frågan om hur Nordens språk kan knytas till DLT.

2. Utbyggbarhet och spridningen

DLT:s översättningsmetod är styrd av två förutsättningar: utbyggbarhetskravet och den idé från vilken beteckningen *Distributed* härrör: spridningen i översättningsprocessen.

Utbyggbarhetskravet gör det nödvändigt att skapa ett väldefinierat interface till vilket godtyckliga utgångs- och målspråk kan knytas, utan att redan befintliga delar av systemet för den skull behöver anpassas. I DLT är detta interface ett **mellanspråk**. DLT:s mellanspråk är en något modifierad version av esperanto. Valet av esperanto

har motiverats i andra arbeten (Witkam 1983; Schubert 1986b, Schubert u.u. b). Orsakerna kan sammanfattas mycket skissartat i en jämförelse med de tre andra typer av teckensystem som skulle kunna tänkas fungera som mellanspråk:

Esperanto lämpar sig för mellanspråksfunktionen i ett datoröversättningssystem bättre än

1. **folkspråk**, eftersom mellanspråket måste vara syntaktiskt oambiguöst, och folkspråken är på språktecknets formsida för oregelbundna;
2. **formella symbolsystem**: Eftersom mellanspråket som enda förbindelselänk mellan utgångs- och målspråk måste återge textens fullständiga innehåll med alla nyanser, är konstgjorda system genom själva sin beskaffenhet otillräckliga (jfr Hjelmlev 1963: 101);
3. **andra planspråk** (volapük, ido, novial, interlingua m fl) eftersom mellanspråket måste äga ett **autonomt** semantiskt system som är oberoende av utgångs- och målspråken. Ett sådant system, som gör ett konstgjort och i början referensspråksberoende system till ett självständigt mänskligt språk, kan inte skapas, utan det kan bara uppstå genom långvarigt oreflekterat bruk av språket i en tillräckligt stor språkgemenskap. Av alla planspråksprojekt har bara esperanto genomgått denna utveckling fullständigt (Blanke 1985: 107ff, särskilt 112 tabell 2; jfr även Bagger 1986: 16ff).

Den andra förutsättningen som karaktäriserar DLT är spridningen. Enligt planerna skall DLT fungera i datakommunikationsnät. Man skall därför inte föreställa sig DLT som en översättningsmaskin som står i ett rum och producerar snyggt tryckta översättningar av texter som matas in i den. DLT kan bäst ses som en software-komponent inom kontorsautomatiseringen som gör befintliga servicetjänster mångspråkiga. Till exempel on-line databankanlitning.

Det är allmänt vedertaget att **helautomatisk** översättning av hög kvalitet är omöjlig. Översättningsdatorm måste alltid få hjälp av en människa. Detta gäller också för DLT. I och med att översättning med mellanspråk på sätt och vis är dubbel översättning, så skulle man egentligen behöva en medhjälpare för utgångsspråket och en för varje målspråk. Men spridningstanken förbjuder detta i praktiken. När man tänker på tillämpningar som databankanlitning och liknande, så är det för dyrt att genast översätta alla texter som matas in i databanken till alla systemets målspråk. Man skulle då vara tvungen att lagra varenda text i så många kopior som man har målspråk, och man skulle dessutom vara tvungen att översätta hela databanken varje gång man lägger till ett nytt målspråk. Därför är det mera praktiskt att översätta varje text med en enda människas hjälp bara till den grad att den kan översättas vidare helautomatiskt så snart en kund vill läsa en bestämd text på ett bestämt målspråk. Hela databanken byggs upp av texter i denna halvöversatta form. Halvvägsprodukten är just en text på mellanspråket. Samma text kan i ett sådant system behöva översättas flera gånger till samma målspråk om olika kunder vid olika tider beställer den. I en sådan spridd uppställning kan man inte ha en medhjälpare för varje målspråksmodul. Därför behövs ett mycket speciellt mellanspråk som möjliggör helautomatisk vidareöversättning. Detta är bland annat vad de tre kriterierna ovan beskriver.

3. Språkspecifikt och språkövergripande

Mellanspråket har fler funktioner än bara att vara en halvvägsprodukt, och dessa funktioner är avgörande för de villkor under vilka man kan knyta flera språk till systemet.

En text på ett godtyckligt språk kan översättas till vilket annat språk som helst. Betraktar man utgångstexten som bestående av språktecken med form och innehåll, så innebär denna tes att det till utgångsspråkets former finns motsvarande former i målspråket med (mer eller mindre) samma innehåll. Detta gäller visserligen inte mellan enstaka ord i två språk, men ganska väl mellan hela texter på de två språken. På så sätt kan man säga att formen är språkspecifik, medan innehållet är språkövergripande. Översättningsprocessen skall förändra formen, men bibehålla innehållet.

Detta är en mycket förenklad framställning. I själva verket är innehållet tyvärr **uppdelat** på olika sätt i olika språks semantiska system (Schubert 1987: 200), så att inte heller innehållet utan vidare kan anses vara den språkoberoende nivå på vilken transfersteget i översättningen kan tas. Det är detta som gör automatisk översättning så svår. Man bör emellertid noggrant fastslå vad svårigheten består i: Problemet är inte att språkoberoende innehåll inte finns, utan att man inte kan skriva upp och hantera rent innehåll på ett språkoberoende sätt. Detta är problemet med betydelse-representationen. Vilket teckensystem man än hittar på för detta ändamål, är det antingen ett mänskligt språk eller också är det beroende av ett sådant. Detta följer av Hjelmslevs (1963: 101) översättbarhetskriterium.

Konsekvensen har för DLT varit att välja själva mellanspråket som betydelse-representation (Schubert 1986c: 146ff). Efter hundra års utveckling, är esperanto på innehållssidan ett fullgott mänskligt språk, men det har kvar den formella regelbundenhet som är typisk för ett planspråk.

En annan konsekvens är ännu väsentligare för anknytningen av nya språk till systemet: Eftersom DLT:s betydelse-representation är mellanspråket, har all semantisk och pragmatisk bearbetning förlagts till mellanspråket. Jag beskriver nedan hur detta går till. Men det är på plats att redan här nämna vad det intressanta med detta tillvägagångssätt är. Lyckas man förskjuta alla processer som har med betydelse att göra till mellanspråket, så behöver man bara utveckla system för dessa tunga och komplicerade processer en enda gång. De kan sedan fungera i systemet för översättningar från och till vilka språk som helst.

Om denna idé är genomförbar så är man framme vid ett ganska smalt och väldefinierat interface till godtyckliga utgångs- och målspråk: En **kontrastiv syntax** och ett **tvåspråkigt lexikon**, båda med mellanspråket som ett av språken.

I stora drag kan översättningsförloppet då beskrivas så här: Först en språkspecifik syntaktisk analys i utgångsspråket. Förekommande alternativ där valet bara kan träffas med hjälp av semantik och pragmatik tas med parallellt till mellanspråket. I systemets kärna utförs de språkövergripande semantiska och pragmatiska bearbetningarna. Till slut följer språkspecifika syntessteg i målspråket. Detta beskrivs mera i detalj i nästa avsnitt.

4. Metatax

Det är omöjligt att beskriva hela översättningsprocessen i ett datoröversättningssystem på ett par sidor. För DLT ges sådana beskrivningar vid olika utvecklingsstadier av systemet av Witkam (1983: III-46ff), Papegaaij (1986: 75ff) och mig (Schubert 1986c: 126ff). Sedan dessa arbeten kom till har DLT förändrats på viktiga punkter. Det nuvarande läget är mera stabilt, eftersom systemet DLT sedan oktober 1987 finns inte bara i lösa moduler, utan också i en sammankopplad preliminär prototyp för översättning från engelska över esperanto till franska, något som i Utrecht i december 1987 uppvisades för fackpressen. Jag skall här försöka ge en snabb genomgång av översättningsprocessen och ta upp särskilt de delar som spelar en roll för anknytningen av nya utgångs- och målspråk.

4.1. Dependenssyntaktisk parsning

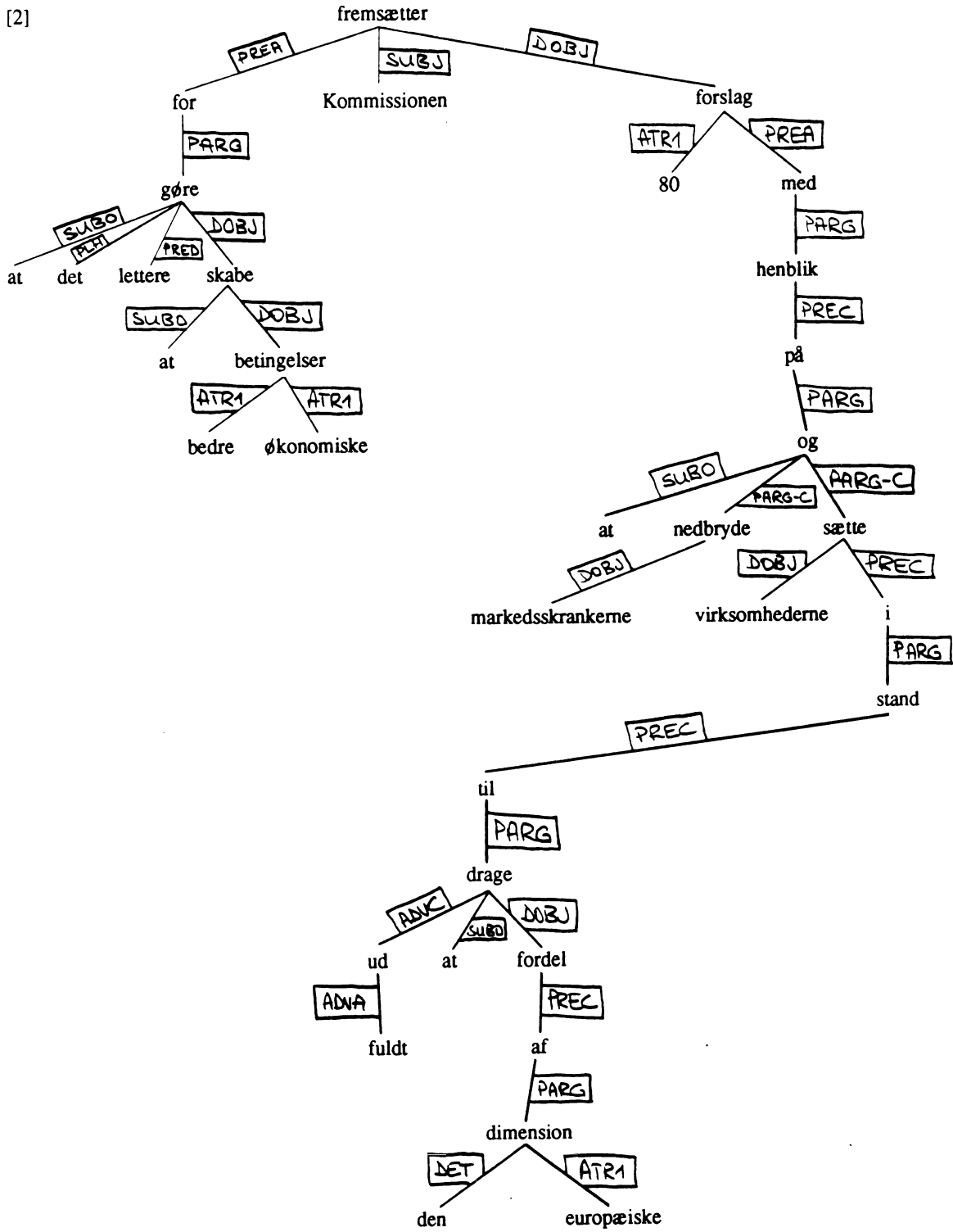
Första steget i DLT-översättningen är en parser som utför den syntaktiska analysen av den inmatade utgångstexten. Inom DLT används en mycket bred definition av begreppet *syntax* som omfattar språktecknets hela formsida. Syntax är alltså formernas grammatik både på ord-, menings- och textnivå. Resultatet av den syntaktiska analysen återges i form av **dependensträd**. Vi har för DLT valt dependenssyntaxen och anpassat den till datalingvistiska behov (se om dependensgrammatiken Tesnière 1959/1982; Nikula 1986; Schubert u.u. a; om DLT:s modell Schubert 1986a: 14ff, 1987: 28ff). Dependensträd bygger inte upp någon abstrakt struktur ovanpå orden, utan har själva orden på noderna. Över varje ord finns en etikett som anger dess syntaktiska funktion i förhållande till det styrande ordet. Ett sådant träd för mening [1] ser ut som [2] på nästa sida.

[1] For at gøre det lettere at skabe bedre økonomiske betingelser, fremsætter Kommissionen 80 forslag med henblik på at nedbryde markedsskranker og sætte virksomhederne i stand til fuldt ud at drage fordel af den europæiske dimension.

Det finns flera orsaker till att föredra dependenssyntax framför konstituensmodeller (Schubert 1987: 193f). Den viktigaste anledningen går ut på att dependensanalysen mera direkt kommer fram till just de drag i textens struktur som är väsentliga för översättningen. Dependenssyntaxen behandlar den syntaktiska **funktionen** primärt och den syntaktiska **formen** sekundärt. I konstituenssyntaxen är det tvärtom.

För att sedan kunna finna orden i ett tvåspråkigt lexikon måste de kunna brytas ned till grundformerna: nominativer, singularisformer, infinitiver osv. Detta kan göras antingen i parsern eller senare med hjälp av redundansregler för lexikonet. I båda fallen måste man anteckna vilka drag ordet hade innan det bröts ned. Men vilka drag måste tas med? Jag diskuterar ovan att man försöker att åtskilja språkspecifika och språkövergripande egenskaper hos ord och ordgrupper och jag talar i det sammanhanget mycket om språktecknet. Det är nyttigt att tänka på tecknet även när det gäller syntaktisk parsning (Schubert 1987: 152ff). Alla syntaktiska (med morfologiska, ordbildningsmässiga m fl) drag är nämligen inte språktecken, dvs de har inte form **och** innehåll. Substantivets numerus har teckenfunktion. Står ett ord i pluralis så översätts det vanligtvis också med ett pluralisord. Men till exempel kasus är

[2]



inte något språktecken. Det är form utan något direkt översättbart innehåll. Kasus pekar på en viss syntaktisk funktion, och den kan i sin tur sedan översättas. Man kan inte gärna beskriva hur en tysk genitiv översätts till svenska utan att ta omvägen över den syntaktiska funktionen. Är genitivordet attribut (*die Last der Schulden*), objekt (*sich der Schulden entledigen*) eller prepositionsargument (*trotz der Schulden*)?

Den syntaktiska analysen levererar ett träd med orden, deras syntaktiska funktioner och översättningsrelevanta drag till nästa steg i processen: den kontrastiva syntaxen. Finns det i utgångsmeningen ambiguiteter som inte kan lösas med enbart syntaktiska medel, så genereras alla syntaktiskt möjliga lösningar parallellt och överförs parallellt till nästa steg.

4.2. Översättningssyntax

Det andra stora bearbetningssteget i DLT-processen är **metataxen**. Ordet är taget från Lucien Tesnières (1959/1982: 283) term *métataxe* för den syntaktiska förändring som utförs i en text eller mening under översättningsprocessen. Metataxen är på samma sätt som parsningen helt formorienterad. Metataxmodulen producerar **alla syntaktiskt möjliga översättningar** till mellanspråket av alla alternativa träd som kom ut ur parsern.

Metataxprocessen kan göras rekursivt, så att antalet kontrastiva syntaktiska regler inte blir oändligt. Formellt sett är hela metataxen trädmanipulation. Man kan bäst bearbeta komplexa träd för hela meningar med bisatser och liknande genom att omforma trädets bit för bit uppifrån och ned. Metataxregler består av ett utgångsspråksträd och ett mellanspråksträd. Trädet för den utgångsspråksmening man vill översätta jämförs med metataxreglernas utgångsspråksträd. Man börjar vid trädets högsta nod (vanligtvis det finita verbet) och söker en metataxregel som i sitt utgångsmönster har den noden, eller en variabel som kan stå för den noden. Har man funnit en sådan regel, så ersätter man den biten av det träd man håller på att omforma med det mellanspråksmönster som ges i regeln. På så sätt uppstår **hybrida träd** som har en **gräns** mellan ord och syntaktiska etiketter från båda språken. Har man utfört en sådan ersättningsoperation, söker man nästa ännu oomformade symbol (ord eller etikett) under gränsen och försöker finna en passande metataxregel.

Här är en finsk mening med sitt träd. [5] är en esperantomotsvarighet till [3]. Metataxen skall omforma [4] till [6]. [7] är ett av de hybrida träd som uppstår halvvägs under metataxprocessen. Givetvis har mening [3] fler översättningsalternativ, bl a ett med betydelsen *li* 'han' för det finska *hän*. Det är en intressant textgrammatisk uppgift att lösa denna pronominala ambiguitet.

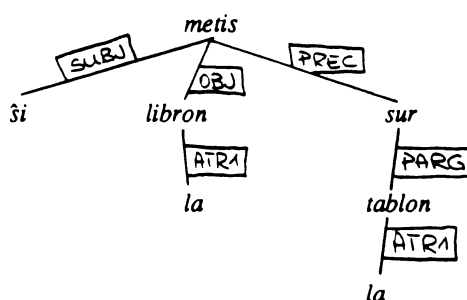
[3] Hän pani kirjan pöydälle.
'han/hon lade boken på-bordet'

[4]

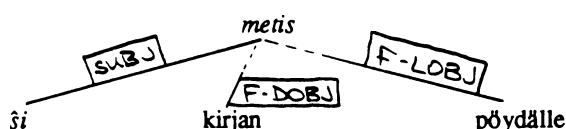


[5] *Ŝi metis la libron sur la tablon.*

[6]



[7]



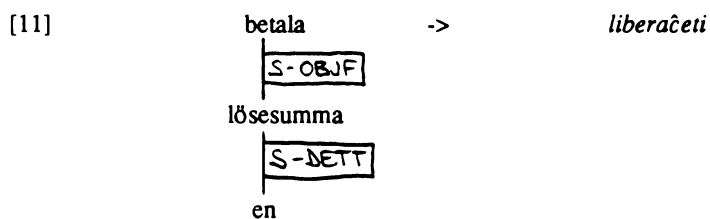
Eftersom metataxen inte är någon urvalsprocess utan skall producera alla syntaktiskt möjliga översättningsalternativ, finns det omarkerade metataxregler som ser till att det alltid finns en regel som passar. Metataxprocessen får aldrig förkasta en mening eller ett översättningsalternativ. Omarkerade regler kan dock inte helt och hållet lösa problemet. Givetvis måste det också finnas ett tvåspråkigt lexikon. Men om ordet finns i lexikonet, så måste det alltid finnas en regel som omformar en syntaktiskt korrekt struktur som innehåller detta ord.

Ett metataxinriktat lexikon har i princip samma struktur som metataxreglerna: Ett träd-diagram i utgångsspråket som omformas till ett mellanspråksträd. I många fall består sådana "träd" i lexikonet bara av en enda nod med ett enda ord på, men som i vanliga ordböcker så är det ofta nödvändigt att översätta från eller till en ordgrupp som enhet. Dessa har då en mera egentlig trädform (exemplen på svenska och esperanto är tagna från Munniksma m fl 1975):

[8] *indexlån -> indeksita prunto*



[10] *betala en lösesumma -> liberaĉeti*



Det som står i lexikonet är transformationsregler för delträd, som utför precis samma

sorts process som metataxreglerna. I princip finns faktiskt ingen skillnad mellan transformationsreglerna i lexikonet och i metataxregelsystemet. Det är fråga om en någorlunda godtycklig gränsdragning mellan de två. Detta beror på att metataxreglerna är **redundansregler** i (eller före) lexikonet. Man skulle teoretiskt kunna ha all transformationsinformation i lexikonet, vilket skulle innebära en ofantlig mängd upprepningar på grund av språkets regelbundenhet. Man har för effektivitetens skull tagit ut ur lexikonet allt som kan formuleras mera generellt i allmännare regler. Dessa är de egentliga metataxreglerna.

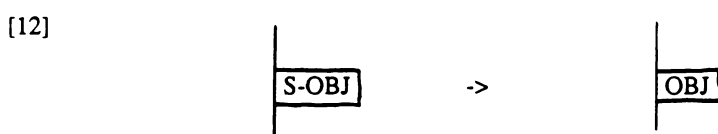
Trots att det kan låta enkelt, är det ganska invecklat att beskriva alla tänkbara syntaktiska omformningar mellan två språk i pålitliga regler. Regelsystemet kan därför bli relativt stort och det är viktigt att inskränka reglernas antal så mycket som möjligt, men att ändå hitta rätt regel snabbt.

För att begränsa regelantalet finns det en effektiv metod: Även om parsern gallrar bort sådana syntaktiska drag som bara identifierar syntaktiska funktioner och som därför inte spelar någon roll i själva översättningen så snart dessa funktioner har gjorts explicita i etiketter på trädets grenar, så kan det ändå finnas alternativa strukturer som har samma översättning. Till exempel stavningsalternativen *da*, *i dag* (två noder) och *idag* (en nod). För att slippa skriva metataxregler för alla sådana alternativ kan man i metataxreglerna införa **utgångsspråksfilter**. Dessa är metataxregler som omformar utgångsspråkliga träd till andra träd i samma språk. Det är alltså ett slags förberedande bearbetning. I exemplet skulle man då omforma *i dag* till *idag* (eller omvänt) och ha en översättningsregel till esperanto bara för den ena formen. Är det då så att sådana formalternativ verkligen betyder exakt samma sak? Finns det inte någon stilskillnad som man vill ha med i översättningen? Metataxmetodiken tillåter att i detta sammanhang återigen behandla sådana skillnader som indikatorer för något annat, men inte som skilda språktecken. Detta innebär att två alternativa former visserligen översätts på samma sätt, men att skillnaden ändå kan ha en funktion, exempelvis som stilmarkör (jfr sv. *ska/skall*, eng. *don't/do not*). Utgångsspråksfiltren kan anteckna detta och ge stilinformationen till vederbörande textgrammatiska stilregler, och ändå översätta båda formerna likadant på meningsnivå.

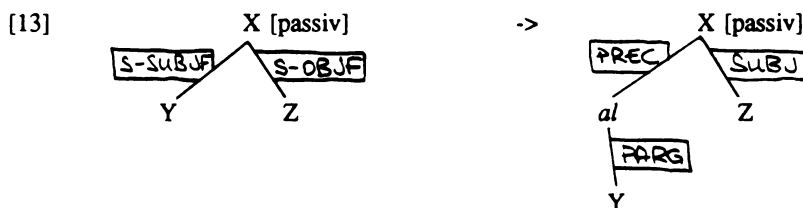
Utgångsspråksfiltren gör det också möjligt att förbereda det träd som skall översättas. Man skiljer ju vanligtvis på innehållsord och funktionsord. Somliga funktionsord översätts normalt inte direkt med ord, utan med syntaktiska drag på vissa innehållsord, till exempel hjälpverb. Man kan därför skriva utgångsspråksfilter även för detta. I stället för två noder för *har kommit* får man då en nod med *komma* och draget "[perfekt]". Hur långt man skall gå med dessa förberedande omformningar beror i princip på ett godtyckligt beslut, men godtyckliga beslut, som är mycket vanliga i grammatiskt systembyggande, skall givetvis vara ändamålsenliga. En "naturlig" gräns till förberedningsomformningarna är lexikonet: För de egentliga transformationsreglerna (alltså för de regler som inte filtrerar inom samma språk utan ersätter träddeklarationer med ord och etiketter från det andra språket) är det viktigt att trädet som översätts och träden i lexikonet förblir jämförbara. Man får alltså genom förberedande filter bygga om trädet bara så mycket att det ännu är jämförbart med de träd som en lexikograf finner naturligt att använda (Schubert 1987: 178).

Jag har redan antytt hur transformationsreglerna fungerar: De bildar hybrida träd

genom att uppifrån och ned söka utgångspråkliga element i trädet och ta fram passande metataxregler för att omforma dessa element till mellanspråket. Dessa transformationsregler kan givetvis inte alltid behandla ett isolerat ord eller en isolerad syntaktisk etikett. Oftast beror valet av en bestämd transformationsregel på ordets omgivning. Hur väljs den rätta regeln ut? Metoden är ganska enkel: Man anger omgivningen i metataxregelns utgångsmönster. Vill man säga att ett svenskt objekt vanligtvis översätts till esperanto som objekt, så behöver man bara en regel som uttrycker detta:



Men vill man uttrycka att objektet till passiva verb översätts enligt en annan regel (*han* [subjekt] *serveras kaffe* [objekt] -> *al li servatas kafo* [subjekt], ordagrant: 'till honom serveras kaffe'), så skall dessa omgivningsvillkor uttryckas i regeln:



Så som reglerna [12] och [13] är formulerade här passar de båda två på passivmeningen *han serveras kaffe*. Ändå används bara [13] utan att detta behöver anges i reglerna. Detta är en praktisk åtgärd som tillåter att formulera allmänna regler som [12], utan att man behöver förändra dem varje gång man skriver en mera speciell regel för något särfall. För att detta skall fungera inför man en **hierarki på metataxreglerna**: Den regel har förtur vars utgångsmönster är mera specifikt. Detta mäts på två sätt: Ett bokstavligt utsatt ord är mera specifikt än en variabel för en hel ordklass, och en struktur med fler noder och etiketter är mera specifik än en sådan med färre element. Det andra villkoret ger regel [13] förtur så att [12] inte används. Det första villkoret ger helt allmänt lexikonreglerna (som har ett ord som högsta nod i sitt utgångsmönster) förtur framför de egentliga metataxreglerna (som har en variabel som högsta nod), så att undantag kan tas om hand i lexikonet. Ibland vill man få fram parallella, strukturellt olika översättningsalternativ. I sådana fall kan man ge en metataxregel statusen "parallell", vilket innebär att också nästa regel i förturshierarkin används.

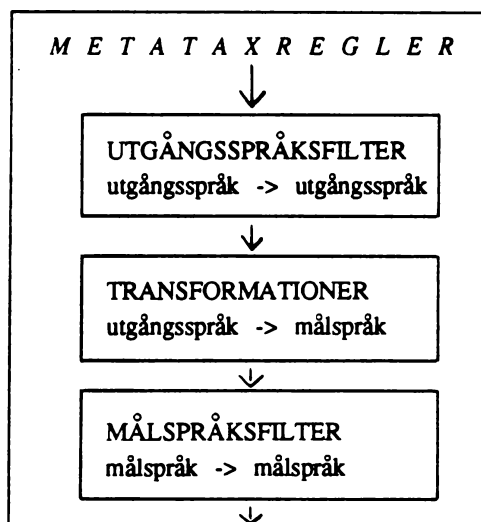
Hierarkin är något man inför i regelsystemet för att hålla det överskådligt och för att kunna bygga ut det på ett hyggligt sätt utan att varje förändring har svårförutsedda biverkningar. En likadan säkerhetsåtgärd är filtreringen **efter** transformationsreglerna. Regelsystemet är trots alla struktureringar så komplext som en kontrastiv syntax brukar vara. Därför är det praktiskt om man kan tillåta vissa övergenerella transformationsregler. Regel [12] till exempel genererar ett objekt under vilket esperantoverb som helst. Men det är ju möjligt att lexikonet någon gång översätter ett transitivt svenskt

verb med ett intransitivt verb på esperanto. För att ändå undvika fel i sådana fall kan man antingen förse varje transformationsregel med ett speciellt korrekhetstest, eller också kan man filtrera bort felen efteråt. Den senare lösningen har valts i DLT:s metataximplementering: Mellanspråksfilter kontrollerar det genom transformationerna genererade trädets korrekthet. Till detta behövs två sorts filter: Det ena kontrollerar om en viss etikett får befinna sig under ett visst ord. Frågan är alltså t ex: Kan det här verbet styra ett objekt? Den andra sortens filter kontrollerar om orden får stå under sina etiketter. T ex: Får en infinitiv vara subjekt? Eftersom metatax inte gör något urval, får underkända konstruktioner inte förkastas utan måste omformas till korrekta träd. Vid detta tillfälle läggs vissa syntaktiska drag till orden, som följer ur de syntaktiska funktionerna. Objekten får sina akkusativändelser osv.

När den egentliga metataxen är färdig så har man ett träd (eller flera alternativa träd) vari alla ord är omsatta till mellanspråket, alla ord står på rätt plats i trädets och vissa syntaktiska drag är utsatta. Nu följer en annan sorts efterfiltrering: reaktions- och kongruensregler. Dessa regler sprider redundanta drag över trädets. Ett objektsubstantivs attributiva adjektiv får nu också sina akkusativändelser osv. Nu föreligger en korrekt mellanspråksmening i trädform. Senare kommer de syntaktiska etiketterna att tas bort och trädets omformas till en mening. Efter denna trädlinearisering får man en vanlig text på mellanspråket.

Metataxsystemet är uppbyggt på följande sätt:

[14]



4.3. Semantik och pragmatik

Innan det sista steget, trädlineariseringen, kan tas måste en enda rätt översättning väljas ut ur mångfalden av syntaktiskt möjliga alternativ, generade av metataxen. Detta sker i DLT genom ett **semantisk-pragmatiskt ordexpertsystem** som bygger på en **kunskapsbank**. Båda delarna är skrivna enbart på mellanspråket esperanto. Eftersom metataxmetodiken innebär att alla semantiska och pragmatiska bearbetningar går ut på

att välja bland redan befintliga alternativa esperantomeningar, så kan samma expertsystem med samma kunskapsbank utföra bearbetningarna oberoende av vad utgångsspråket var. Detta är givetvis bara möjligt om mellanspråket är autonomt och inte tillyxat efter vissa utgångsspråks modeller (jfr avsnitt 2). Jag tar här inte upp detaljer om ordexpertsystemet. Det har beskrivits mycket utförligt av Papegaaij (1986: 75ff).

Efter det semantisk-pragmatiska urvalet kan tveksamma alternativ kvarstå. Kan systemet inte med tillräcklig säkerhet komma fram till en enda översättning, så rådfrågas människan i en interaktiv dialog. Här parafraseras de kvarstående alternativen på utgångsspråket, så att den som arbetar med DLT varken behöver vara översättare eller behöver behärska vare sig esperanto, målspråken eller dependensgrammatiken. DLT är ett användarsystem, inte någon översättarhjälp.

4.4. Från mellanspråk till målspråk

Har ordexpertsystemet, eventuellt med människans hjälp, valt ut den riktiga översättningen, kan trädet lineariseras och den första hälften av översättningsprocessen är färdig. Texten kan nu läsas på esperanto. Den kodas för nätöverföringen och mottas i denna form av en mottagarmodul. Där avkodas den och analyseras i en parser, vilket går snabbt eftersom DLT:s mellanspråk är syntaktiskt oambiguöst. Här efter följer metataxen, den här gången från mellanspråket till målspråket.

Den enda väsentliga skillnaden mot första hälften av processen ligger i den semantiska bearbetningen. Här används samma ordexpertsystem för att välja ut den bästa översättningen i målspråket. Hur kan det vara möjligt när ordexpertsystemet är gjort för esperanto och inte för målspråket? Den lösning som möjliggör detta bygger på en idé som är mycket vanlig i normala ordböcker för mänskligt bruk: Har ett utgångsspråksord flera översättningar så anges i ordböcker ofta i vilken kontext vilken av översättningarna skall väljas. Denna kontextuppgift ges normalt på utgångsspråket. På samma sätt fungerar de tvåspråkiga lexikonerna för målspråk inom DLT.

Detta är ett exempel på en ingång i DLT:s esperanto-franska lexikon. Först står utgångsordet (esperanto, med morfemgränser angivna), sedan en semantisk relator (ett esperantomorfem) och sedan en rad esperantoord som anger en typisk omgivning i vilken det franska målordet skall användas som ges i sista spalt. Där under ger jag ungefärliga svenska glosor (*akr'a* betyder bl a 'skarp').

[15]

'*akr'a*', ['a', '*dolor'o*, *mal'varm'o*, *riproĉ'o'j*, *vort'o'j*, *romp'o*, *eĝ'o*'], ['*viŝ'*/'ADJ']

smärta, köld, förebråelser, ord, brytning, kant

'*akr'a*', ['a', '*naz'o*, *orel'o'j*, *tur'o*'], ['*pointu'*/'ADJ']

näsa, öron, torn

'*akr'a*', ['a', '*spic'o*, *pipr'o*, *brand'o*'], ['*fort'*/'ADJ']

krydda, peppar, brännvin

'*akr'a*', ['a', '*disput'o*, *batal'o*, *kriz'o*, *vent'o*, *ŝtorm'o*'], ['*violent'*/'ADJ']

dispyt, strid, kris, vind, storm

'*akr'a*', ['a', '*ironi'o*'], ['*mordant'*/'ADJ']

ironi

...

På samma sätt som ordexpertsystemet i översättningsprocessens första hälft väljer ut

det i kontexten mest sannolika alternativet, så gör det det också nu, nämligen genom att jämföra olika alternativs kontextförväntningar med den aktuella kontexten i den text som översätts. Dessa jämförelser utförs i esperantotexten som föreligger fullständig och i en av människan i dialogen disambiguerad och godkänd form.

Har den rätta översättningen på så sätt valts ut, så kan den lineariseras till en vanlig text och översättningen är fullbordad.

5. Hur knyter man fler språk till systemet?

Den skissartade genomgången av översättningsförloppet i systemet DLT visar i grova drag DLT:s modulära struktur. Vad måste nu till för att knyta ett nytt utgångs- eller målspråk till detta system?

Det är ett enda regelsystem som knyter ett språk till DLT:s kärna: metataxen. Detta system omfattar de egentliga metataxreglerna och ett tvåspråkigt lexikon med mellanspråket. En metatax förutsätter en dependenssyntax för språket i fråga (och för mellanspråket). Metataxen med lexikonet behövs vare sig det nya språket skall vara utgångs- eller målspråk, men lexikonet är något olika i de två fallen. För att kunna tillämpa ett metataxsystem på ett utgångsspråk måste det dessutom finnas en parser som analyserar den inmatade texten enligt samma dependenssyntax som metataxreglerna bygger på, och en dialogmodul. För ett nytt målspråk krävs givetvis ingen parser, men däremot trädlineariseringsregler. I båda fallen behövs för ett nytt språk först en dependenssyntax med ett syntaktiskt lexikon och sedan en metatax med ett tvåspråkigt lexikon.

De ovan nämnda förstudierna för anknytningen av fler språk till DLT gäller då i första hand dependenssyntaxer och sedan metataxer. Vad gäller Nordens språk så har dependenssyntaxer enligt DLT-modell utarbetats (med ännu inte publicerats) för finska av Kalevi Tarvainen (1987) och för danska av Ingrid Schubert (1987). En isländsk arbetsgrupp har just bildats. Intressenter för övriga språk är välkomna. Från språkområden utanför Norden har DLT redan fått en tysk och en ungersk dependenssyntax (Lobin 1987; Prószyky/Koutny/Wacha 1987), medan arbetsgrupper på olika håll har börjat arbeta med en rad romanska, slaviska och östasiatiska språk. Flera väntas tillkomma under 1988.

Noter

1. Träddiagrammen i denna uppsats är tänkta som illustrationer. Givetvis kan jag här inte förklara i detalj de bakomliggande syntaxbeskrivningarna. Jag ber läsaren om ursäkt för att jag citerat opublicerat material om finskan och danskan. Angående finskan har jag också lånat en exempelmening från Tarvainen (1985: 166). De svenska träden är mina egna preliminära förslag, och esperantoträden motsvarar DLT:s mellanspråkssyntax (Schubert 1986: 23ff).
2. Jag tackar Lena Odén för språkgranskningen.

LITTERATUR

- Bagger, Preben (1986): *Sprog og sprog imellem*.
Skelby: Kommunikation og Kultur
- Blanke, Detlev (1985): *Internationale Plansprachen*.
Berlin: Akademie-Verlag
- Hjelmslev, Louis (1963): *Sproget*.
København: Berlingske forlag (2:a uppl.)
- Lobin, Henning (1987): *Dependenzsyntax des Deutschen*.
Opublicerad rapport. Utrecht: BSO/Research
- Munniksma, F. m fl (1975): *International business dictionary in nine languages // Internacia komerca-ekonomika vortaro en naŭ lingvoj*.
Deventer / Antwerp: Kluwer
- Nikula, Henrik (1986): *Dependensgrammatik*.
Malmö: Liber
- Papegaaij, B. C. (1986): *Word expert semantics. An interlingual knowledge-based approach*.
Utg. V. Sadler / A. P. M. Witkam. Dordrecht / Riverton: Foris
- Prószéký, Gábor / Ilona Koutny / Balázs Wacha (1987): *A dependency syntax of Hungarian (for use in DLT)*.
Opublicerad rapport. Utrecht: BSO/Research
- Schubert, Ingrid (1987): *Dänische Dependenzsyntax für DLT*.
Opublicerad rapport. Utrecht: BSO/Research
- Schubert, Klaus (1986a): *Syntactic tree structures in DLT*.
Utrecht: BSO/Research
- Schubert, Klaus (1986b): *Wo die Syntax im Wörterbuch steht. Esperanto als Brückensprache der maschinellen Übersetzung*.
i: *Pragmantax*. Utg. Armin Burkhardt / Karl-Hermann Körner. Tübingen: Niemeyer, s. 449-458
- Schubert, Klaus (1986c): *Linguistic and extra-linguistic knowledge*.
i: *Computers and Translation* 1, s. 125-152
- Schubert, Klaus (1987): *Metataxis. Contrastive dependency syntax for machine translation*.
Dordrecht / Providence: Foris
- Schubert, Klaus (under utgivning a): *Inbjudan till en tillämplig språkteori*.
i: *Nysvenska studier*
- Schubert, Klaus (under utgivning b): *Ausdruckskraft und Regelmäßigkeit. Was Esperanto für automatische Übersetzung geeignet macht*.
i: *Language Problems and Language Planning*
- Tarvainen, Kalevi (1985): *Kontrastive Syntax Deutsch - Finnisch*.
Heidelberg: Groos
- Tarvainen, Kalevi (1987): *Formale Dependenzgrammatik des Finnischen*.
Opublicerad rapport. Utrecht: BSO/Research
- Tesnière, Lucien (1959/1982): *Éléments de syntaxe structurale*.
Paris: Klincksieck (2:a uppl., 4:e tryck. 1982)
- Witkam, A. P. M. (1983): *Distributed Language Translation. Feasibility study of a multilingual facility for videotex information networks*.
Utrecht: BSO