

Hans Broekhuis. *Derivations and Evaluations: Object Shift in the Germanic Languages.* Berlijn/New York: Mouton de Gruyter, 2008 (Studies in Generative Grammar 97). ISBN 978-3-11-019864-5 (HC). xiv + 382 pagina's. EUR 98,00.

Er zijn ruwweg twee manieren om onderzoek te doen. Ofwel je kiest een onderwerp en probeert daar een goed idee over te ontwikkelen, ofwel je hebt een goed idee en zoekt een onderwerp waarop je het vruchtbaar kunt toepassen. Hans Broekhuis, ook bekend van het grote project *Modern Grammar of Dutch*, is iemand die in zijn theoretische werk consequent kiest voor de tweede strategie. Het begon met zijn proefschrift *Chain Government: Issues in Dutch Syntax* (1992), en nu is daar het ambitieuze *Derivations and Evaluations*. Het centrale idee is een nieuwe organisatie van de grammatica, zo dat belangrijke aspecten van het generatieve minimalistische programma gecombineerd worden met de optimaliteitstheorie (OT). Het concrete onderwerp van studie is de positionering van argumenten en werkwoorden in de Germaanse talen, waarbij *object shift* (waaronder in dit boek ook *scrambling* verstaan wordt) als uitgangskwestie dient.

Broekhuis introduceert het Derivaties en Evaluaties-framework met de constatering dat alle Chomskyaanse grammaticamodellen sinds de jaren zeventig bestaan uit een generatieve component en een filtercomponent. De eerste zorgt voor de opbouw van syntactische structuur (denk aan Merge of Move α), de tweede zorgt ervoor dat ongrammaticale structuren worden uitgefilterd door middel van condities op representaties (denk aan het *that*-trace filter of conditie A van de bindingstheorie). In het huidige minimalistische programma lijkt de aandacht wat eenzijdig gericht op de generatieve component (*Derivatie*), maar in de praktijk wordt er natuurlijk lustig op los gefilterd door middel van sterke en zwakke features, PF-deletiemechanismes, enzovoort. In de concurrerende OT-syntaxis is het filteren echter de kern van de zaak (*Evaluatie*); minder aandacht is er voor de vraag hoe de te evalueren representaties (de 'kandidaten') gegenereerd zijn. Volgens Broekhuis hoeven we deze twee modellen niet als rivaliserend te zien: ze kunnen elkaar juist versterken als ze in elkaar worden geschoven. Beide hebben immers een sterke en een zwakke kant, die complementair zijn. Dat levert het volgende plaatje op (mijn weergave):

(1) input $\rightarrow$ generator: $MP_{\text{kern}} \rightarrow$ kandidatenlijst $\rightarrow$ evaluator: $OT_{\text{kern}} \rightarrow$ optimale output

De *input* komt uit het lexicon. De *generator* is de kern van het minimalistisch programma, bestaande uit – als ik het goed begrijp – de operaties Select (Merge), Move, Agree en mogelijke beperkingen daarop, bijvoorbeeld lokaliteitseffecten als gevolg van fases (generatieve cycli) en de Last Resort Condition, die zegt dat verplaatsing zonder trigger (dus feature checking) niet mogelijk is. Alleen volledig interpreteerbare structuren worden doorgelaten (Full Interpretation); dat levert dan een *kandidatenlijst* met relevante syntactische structuren op. Die worden middels de OT-methode *geëvalueerd* (waarover zo dadelijk meer), wat een *optimaal* – doorgaans uniek – resultaat oplevert. Het geheel onder (1) mogen we 'core syntax' noemen, wat een welkome correctie is op de lijn van Chomsky, die de laatste jaren steeds meer onderdelen uit de syntaxis verbaant naar een niet nader uitgewerkte periferie.

Het Derivaties en Evaluaties-model is niet geheel nieuw. Het werd eind jaren negentig ontwikkeld in samenwerking met Joost Dekkers aan de Universiteit van Amsterdam, waar ik zelf overigens getuige van was. De eerste resultaten hadden betrekking op het COMP-domein van betrekkelijke bijzinnen. Het proefschrift van Dekkers (1999), met gelijklopende boventitel (*Derivations and Evaluations*), gaat over subjecten en voegwoorden in het Frans en Nederlands. Ondanks de LOT/AVT/Anéla-dissertatieprijs (moet die naam niet eens verkort worden?) is

Dekkers uit de taalkunde gestapt, maar zijn co-promotor Hans Broekhuis is verdergegaan op de ingeslagen weg, met als resultaat de huidige gedetailleerde analyse van object shift en verwante zaken.

Een centrale rol in het boek speelt tabel 1 op blz. 120, en meermaals herhaald, waarin gecomprimeerd de eigenschappen van object shift in acht Germaanse talen zijn weergegeven. Voor het idee is deze hier gereproduceerd, al kunnen de details niet in enkele woorden worden besproken:

Table 1: *Restrictions on the application of regular object shift*

	weak pronouns	DPs	across P	across IO	across V	precedes non-verbal particles
Finnish-Swedish	—	—	n.a.	n.a.	n.a.	n.a.
English	—	—	n.a.	n.a.	n.a.	n.a.
Swedish	optional	—	—	—	—	—
Norwegian	optional	—	—	—	—	+
Danish	obligatory	—	—	—	—	+
Icelandic	obligatory	optional	—	—	—	+
Dutch	obligatory	optional	—	—	+	+
German	obligatory	optional	—	+	+	+

Een eenvoudig Nederlands voorbeeld is (2). In de canonieke volgorde staat een object (hier: *dit boek*) in de werkwoordelijke groep na een eventueel zinsadverbium (hier: *waarschijnlijk*); als object shift plaatsvindt komt *dit boek* vóór *waarschijnlijk* te staan.

(2) Jan heeft <dit boek> waarschijnlijk <dit boek> gelezen.

De keuze hangt af van de vraag of het object tot de presuppositie dan wel de informatiefocus van de zin behoort. In het eerste geval is een ‘shift’ nodig, in het tweede geval blijft het object in de werkwoordelijke groep staan. Dit verklaart meteen waarom voornaamwoorden, die geen nieuwe informatie brengen, normaalgesproken verplicht naar voren worden gehaald: *Jan heeft <het> waarschijnlijk *<het> gelezen.*

Niet alle talen doen aan object shift. Broekhuis leidt een macroparametrisatie op dit vlak af uit de interactie van twee constraints. *MOVE is op het eerste gezicht een van de simpelste en meest algemene constraints: het straft verplaatsing af; EPP(case) zegt dat een niet-gevalueerd naamvalskenmerk zijn ‘goal’ aantrekt. Dat laatste vereist enige achtergrondkennis. Een EPP-constraint heeft de uitwerking van een sterk feature, wat wil zeggen dat de relatie tussen twee grammaticale elementen strikt lokaal tot stand moet worden gebracht. In dit geval vindt er agreement plaats tussen het object (dat gegenereerd wordt als zuster van het hoofdwerkwoord en dat, zoals elke naamwoordgroep, abstracte casus draagt) en het functionele hoofd *v* (dat het werkwoord transitief maakt, dus het externe argument oproept, en tegelijk de accusatief licenseert, in overeenstemming met Burzio’s generalisatie – zie Burzio 1986). Als EPP(case) nu hoger geordend is dan *MOVE, dan wordt het object om die reden verplaatst naar het middenveld (Spec*v*P); omgekeerd, als geldt *MOVE > EPP(case), dan kan er geen verplaatsing zijn. Beide mogelijkheden worden gegenereerd en zijn onderdeel van de kandidatenlijst, maar de OT-evaluator bepaalt – afhankelijk van de taalspecifieke constraint-ordening – welke van de

verschillende mogelijkheden als optimale output tevoorschijn komt. De talen met volledige object shift zijn het IJslands, Duits en Nederlands. Talen zonder object shift zijn bijvoorbeeld het Engels en Frans. In de Scandinavische talen (Deens, Noors, Zweeds) ligt het gecompliceerder: er is alleen sprake van ‘pronoun shift’ onder bepaalde omstandigheden.

OT-constraints zijn universeel en hebben een algemeen karakter. In principe is elk constraint schendbaar, wat samenhangt met het idee dat er allerlei tegengestelde krachten werkzaam zijn in de taal, net als in het bovenstaande voorbeeld. OT is meteen ook een theorie over taalvariatie: de verschillende mogelijke ordeningen tussen de constraints bepalen het aantal mogelijke talen. Broekhuis werkt nauwkeurig uit welke constraints nodig zijn voor een analyse van object shift en hoe deze op elkaar inwerken in de verschillende Germaanse talen. Empirische factoren die hierbij een rol spelen zijn de positie van het werkwoord (denk aan Holmbergs generalisatie, waarover straks meer), presuppositie en focus, en de eventuele aanwezigheid van hulpwerkwoorden, partikels, indirect objecten, bijwoorden en bijwoordelijke bepalingen. Het geheel is behoorlijk complex en veelomvattend, maar de tekst behandelt stap voor stap alle afleidingen en mogelijke complicaties. Daardoor is die soms wat droog, maar Broekhuis verdient mijns inziens grote waardering voor zijn gedetailleerde bespreking.

Het boek bevat een handige appendix waarin alle gebruikte constraints van de evaluator kort worden besproken. Er is geen appendix waarin de componenten van de generator worden samengevat. Dat is eigenlijk jammer: de titel wekt de verwachting van een meer evenwichtige benadering van deze twee modules, maar de nadruk blijkt in de praktijk te liggen op de OT-kant van de analyse. Het Derivaties en Evaluaties-model roept bij mij een aantal vragen op; deze betreffen met name de interactie tussen de verschillende onderdelen van het taalsysteem. Laat ik drie zaken noemen.

1. Broekhuis stelt terecht dat het wenselijk is dat het aantal kandidaten per evaluatie eindig en beperkt is. De specifieke MP-generator zorgt daarvoor. In navolging van Grimshaw (1997) neemt hij aan dat alleen kandidaten met een gelijke betekenis in dezelfde kandidatenset kunnen zitten. De aanname van een zogenaamde numeratie (een geselecteerde set van lexicale items bij aanvang van de derivatie), waartegen een aantal bezwaren wordt aangevoerd, is dan ook overbodig. Maar de vraag is waar de betekenis van een representatie wordt bepaald. De standaardaanname is dat de syntaxis volledige representaties doorgeeft aan de PF- en LF-interface, de toegangspoorten tot respectievelijk de uitspraak en de interpretatie. In het Derivaties en Evaluaties-model, zou je in eerste instantie denken, wordt de uiteindelijke optimale output naar de interfaces gestuurd. Maar dan zou de betekenis (en uitspraak) na afloop van de evaluatie van de kandidatenset komen, terwijl die set juist moest worden vastgesteld aan de hand van gelijke betekenissen: een onmogelijk dilemma. Een uitweg is dat de interface zich vóór de evaluator bevindt, zodat elke *potentiële* kandidaat gewogen kan worden. Behalve in de theorie van fases vindt dit idee bevestiging in de eis van Full Interpretation: alle features van een kandidaat moeten interpreteerbaar en dus gevalueerd/gecheckt zijn. Ontbreekt er dan niet iets in de weergave van het model in (1)? En wat gebeurt er na de evaluatie van de kandidaten met de optimale output – moet die opnieuw naar de interfaces? Wat is dan de status van de zogenaamde LF- en PF-constraints? Meer fundamentele discussie en uitleg was volgens mij op zijn plaats geweest.

2. Constraints zijn per definitie filters op *representaties*. Ik verwacht dan dat ze iets kunnen zeggen over het resultaat van een mogelijke bewerking. Sommige van de voorgestelde constraints verwijzen echter expliciet naar de voorafgaande *derivatie*. Het genoemde *MOVE is hier een duidelijk voorbeeld van: het wordt gedefinieerd als “don’t apply internal merge” (blz. 37 en 355). Maar ‘internal merge’ is een proces van de generator; het is niet rechtstreeks zichtbaar voor de

evaluator. Uiteraard kunnen we aannemen dat dit proces traceerbare sporen achterlaat in de resulterende representatie. Een representationeel evalueerbaar constraint zou er dan zo uit kunnen zien: $*[... X(P) ... t_{x(p)} ...]$, in woorden: straf sporen van verplaatsing af. Broekhuis' definitie "don't apply internal merge" moet volgens mij gezien worden als een heuristiek op basis van een dergelijk filter. Het gevaar daarvan is dat een deel van de theoretische last die er in feite mee gepaard gaat aan het zicht onttrokken wordt. Voor het filter *Move, bijvoorbeeld, is niet alleen het proces internal merge van belang, maar ook het bestaan van sporen (of eigenlijk kopieën, binnen het minimalistisch programma); dat laatste is overigens tamelijk standaard, maar niet geheel onomstreden.

3. Broekhuis gaat uit van een universele *specificeerder–hoofd–complement*-structuur. Op blz. 73-96 laat hij behoorlijk overtuigend zien dat een VO-basis zelfs in het Nederlands met zijn ogenschijnlijke OV-volgorde (modulo V2) theoretisch gunstiger uitpakt dan een OV-basis, als we tenminste niet alleen naar simpele zinnen kijken, maar ook rekening houden met werkwoordsclusters en de positie van partikels en secundaire predicaten. Dat is een mooie bevestiging van eerder werk van Kayne (1994) en Zwart (1997). Maar hoe en waar moet *spec-hoofd-compl* worden afgedwongen? In de appendix wordt gesuggereerd dat er naast H-COMPL een OT-constraint SPEC-H bestaat. Maar die zijn dan in principe schendbaar – dat kan toch niet de bedoeling zijn, zou je zeggen. In het boek worden ook nergens structuren geëvalueerd die in dit opzicht 'verkeerd' zijn (bv. met een OV- of VS-basis). Bovendien wordt H-COMPL als 'order preservation constraint' getypeerd, en dat lijkt mij niet hetzelfde: "[it] favor[s] preservation of [...] the underlying head-complement order" (blz. 357). Maar dan moet de onderliggende volgorde dus op een andere manier vastgesteld worden, anders valt er niets te behouden. De uiteindelijke definitie van H-COMPL luidt echter nog anders: "a head precedes all terminals originally dominated by its complement". Hier is geen sprake van 'preservation', alleen van een gewenste uiteindelijke volgorde (ook na eventuele verplaatsingen). Het zou aan mij kunnen liggen, maar als lezer ben ik nu de draad kwijt welke kant het op moet met de universele basisstructuur. Hoe dan ook, met behulp van H-COMPL kan Holmbergs generalisatie – zie Holmberg (1986) – worden afgeleid: bij object shift moet ook het werkwoord verplaatst worden, anders zou het complement voorafgaan aan het hoofd. Dit proces is van kracht in het Zweeds, maar niet in het Nederlands, zodat bij ons H-COMPL laaggerankt is (ten gunste van andere, in het Nederlands belangrijkere constraints). Daaruit volgt dat na verplaatsing de volgorde *compl (...)* *hoofd* kan ontstaan. Over de gegenereerde basis zegt het denk ik niets. Merk op dat, aangezien er toch verplaatst wordt, het voor de uiteindelijke volgorde niet uitmaakt of we in eerste instantie *compl-hoofd* of *hoofd-complement* genereren – althans in dit ene geval.

Laat ik ten slotte kort iets zeggen over de indeling van het boek. Er zijn vijf inhoudelijke hoofdstukken, plus een korte samenvatting, een appendix met constraints, een literatuurlijst en een beknopte index op onderwerp. Het boek is uiterlijk mooi verzorgd. Een minpuntje is het gebrek aan hiërarchie in de inhoudsopgave, wat gezien de vele sub(sub(sub))-paragrafen geen overbodige luxe was geweest. Hoofdstuk 1 bespreekt het Derivaties en Evaluaties-framework. In hoofdstuk 2 betoogt Broekhuis dat er in de literatuur ten onrechte verschil is gemaakt tussen Scandinavische object shift en Ned./Duitse scrambling; beide betreffen A-verplaatsing. Vervolgens laat hij zien dat we wel onderscheid moeten maken tussen gewone object shift (inclusief scrambling) en korte object shift (binnen de VP). Dit laatste vindt plaats in alle Germaanse talen (inclusief het Engels). Een en ander hangt samen met de discussie over VO/OV. Overigens wordt zogenaamde 'medium scrambling' zoals O-over-S in het Duits in dit boek buiten beschouwing gelaten; hoewel de afbakening iets duidelijker had gekund, lijkt me dat

verdedigbaar omdat dit soort scrambling van een ander type is, namelijk A-bar-verplaatsing, waarschijnlijk gerelateerd aan focus. Hoofdstuk 3 gaat over gewone object shift. Hierin worden de mogelijkheden en beperkingen met betrekking tot object shift besproken, en de verschillen tussen de individuele Germaanse talen. Het OT-systeem doet ook voorspellingen ten aanzien van het aantal mogelijke verschillende talen. Hoofdstuk 4 bespreekt de interactie tussen object shift en een aantal andere soorten verplaatsingen, waaronder A-bar-verplaatsing zoals bij topicalisatie, vragen en focusconstructies. Een van de mooiste delen uit het boek vind ik paragraaf 4.2, waarin de parallellen tussen object shift en subject shift worden besproken. Een voorbeeld van het laatste is *Morgen komt <mijn vader> waarschijnlijk <mijn vader> op bezoek*. Hoofdstuk 5 gaat over predicaatsverplaatsing. Dit had een boek op zichzelf kunnen zijn. Broekhuis beoogt hiermee te laten zien dat het geïntroduceerde systeem, in combinatie met de verschillende achtergrondaannames, ook in gunstige zin op andere constructies dan object shift toegepast kan worden. De discussie betreft onder meer inversie en resultatieve small clauses in het Engels en Nederlands.

Kortom, *Derivations and Evaluations* is van belang voor iedereen die geïnteresseerd is in de opbouw van het middenveld in de Germaanse talen. Meer in het algemeen is het eerste hoofdstuk een waardevolle bijdrage aan de discussie over de architectuur van de grammatica.

Verwijzingen

- Broekhuis, Hans. 1992. *Chain Government: Issues in Dutch Syntax*. Proefschrift, Universiteit van Amsterdam. (HIL dissertaties 2.)
- Burzio, Luigi. 1986. *Italian Syntax: A Government-Binding Approach*. Dordrecht: Reidel.
- Dekkers, Joost. 1999. *Derivations and Evaluations: On the Syntax of Subjects and Complementizers*. Proefschrift, Universiteit van Amsterdam. (LOT dissertaties 21.)
- Grimshaw, Jane. 1997. Projection, heads and optimality. *Linguistic Inquiry* 28: 373-422.
- Holmberg, Anders. 1986. *Word Order and Syntactic Features in the Scandinavian Languages and English*. Proefschrift, Universiteit van Stockholm.
- Kayne, Richard. 1994. *The Antisymmetry of Syntax*. Cambridge MA: MIT Press.
- Zwart, Jan-Wouter. 1997. *Morphosyntax of Verb Movement: A Minimalist Approach to the Syntax of Dutch*. Dordrecht: Kluwer.

Mark de Vries

Algemene Taalwetenschap, Rijksuniversiteit Groningen
mark.de.vries@rug.nl