

C-COMMANDEREN EN DE CONFIGURATIONELE MATRIX

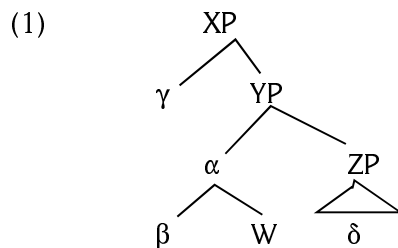
Reactie op 'De primaire structuur' van Jan Koster¹

Jan-Wouter Zwart

In de vorige aflevering van *TABU* heeft Jan Koster betoogd dat de derivationele afleiding van de definitie van c-commanderen, voorgesteld in Epstein *et al.* (1998:32), oude wijn in lelijke nieuwe zakken is.² C-commanderen is volgens Koster een normale afhankelijkheidsrelatie, die gedefinieerd is volgens de principes van de configurationele matrix (Koster 1987:8). Epstein stelt dat de definitie van c-commanderen te begrijpen is als we er van uit gaan dat syntactische structuren stap voor stap worden opgebouwd, door telkens twee constituenten met elkaar te verenigen ('Merge'; vgl. Chomsky 1995:243). Koster stelt daarentegen dat ook Epstein's definitiewijze gebruik maakt van de specifieke eigenschappen van syntactische structuur, en dat die eigenschappen weer volgen uit de configurationele matrix.

In deze reactie wil ik het opnemen voor de theorie van Epstein. Volgens mij doet deze theorie wat hij moet doen, namelijk antwoord geven op een fundamentele, tot voor kort onbeantwoorde en zelfs ongestelde vraag: waarom is c-commanderen gedefinieerd zoals het gedefinieerd is? Waarom niet geheel anders?

De definitie van c-commanderen is het beste aanschouwelijk te maken in een boomstructuur. In (1) c-commandeert α δ :



Maar α c-commandeert γ niet, en β c-commandeert δ niet. Eén ander is neergelegd in de vertrouwde definitie van c-commanderen, volgens welke α δ alleen dan c-commandeert als de eerste vertakkende knoop die α domineert (YP) ook δ domineert. (YP domineert γ niet, dus α c-commandeert γ niet, en de eerste vertakkende knoop die β domineert is α en α domineert δ niet, zodat β δ niet c-commandeert.)

C-commanderen is zo gedefinieerd omdat de observaties daartoe dwingen. Afhankelijkheidsrelaties (bijvoorbeeld binding van anaforen) komen wel voor tussen elementen in de posities α en δ in (1), maar niet tussen elementen in de posities α en γ of β en δ in (1). Dus, naast (2a), met $\alpha = \textit{Jan}$ en $\delta = \textit{zichzelf}$, kunnen we niet hebben (2b), met $\alpha = \textit{Jan}$ en $\gamma = \textit{zichzelf}$, of (2c), met $\beta = \textit{Jan}$ en $\delta = \textit{zichzelf}$.

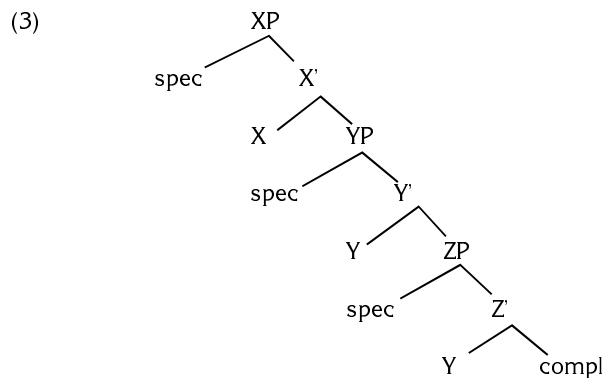
- (2) a. ..dat *Jan zichzelf zag*
b. * ..dat *zichzelf Jan zag*
c. * ..dat *Jan z'n ouders zichzelf zagen*

Over de definitie van c-commanderen bestaat geen controverse, en ook niet over de syntactische structuren waar de c-commandeerrelatie op gedefinieerd is. Laten we er voor het gemak van uit gaan dat dat altijd binaire, aan de rechterkant vertakkende structuren zijn (zoals Koster en Epstein

¹ Koster (1999).

² In de rest van dit stukje verwijs ik naar Epstein *et al*/met 'Epstein'.

allebei doen) (*spec* = specificierder, *compl* = complement):

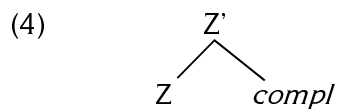


Voor Koster is de c-commandeerrelatie een normale afhankelijkheidsrelatie: we hebben een afhankelijk element (δ in (2)) en een antecedent (α in (2)) die samen in één lokaal domein zitten (YP in (2)). De keus van het lokale domein is in Koster's theorie van afhankelijkheidsrelaties variabel. In dit geval valt het lokale domein samen met de eerste vertakkende knoop. Dit laat de vraag open waarom het lokale domein in het geval van de c-commandeerrelatie samenvalt met de eerste vertakkende knoop. Het is deze vraag die Epstein wil beantwoorden.

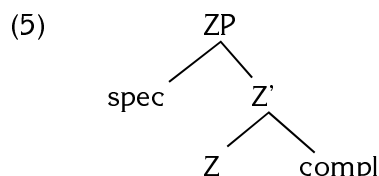
Epstein stelt de voor de geroutineerde taalkundige wat verrassende vraag waarom de c-commandeerrelatie precies zo is als hij is. Met andere woorden, waarom in (3) Y wel alles c-commandeert wat gedomineerd wordt door Y', maar niet bijvoorbeeld X of de specifier van XP. Dat is geen zuiver technische vraag, want in feite wordt hier gevraagd naar de reden dat X (in (3)) nooit het afhankelijke element kan zijn in een relatie met Y (dus, mutatis mutandis, naar de reden van de ongrammaticaliteit van (2c)). Koster sluit dergelijke afhankelijkheidsrelaties uit door de extra eis op te nemen dat het antecedent aan het afhankelijke element moet voorafgaan. Als syntactische structuren er altijd uit zien als in (3) volgt daaruit onmiddellijk dat de c-commandeerrelatie begrensd wordt door de eerste vertakkende knoop. Op de wenselijkheid van zo'n extra eis kom ik straks nog terug.

De vraag naar de definitie van c-commanderen is in wezen een vraag over ongelijkheid: waarom c-commandeert α de ene knoop wel en de andere knoop niet? Deze vraag hoeft niet gesteld te worden als α alle knopen in een structuur c-commandeert. In (3) bijvoorbeeld c-commandeert de hoogste specifier alle andere knopen in de boom, en hoeft de vraag naar de definitie van c-commanderen niet gesteld te worden. C-commanderen is in dat geval een *totale relatie* tussen het antecedent en alle andere knopen in de boom.

Van dat gegeven maakt Epstein gebruik om de definitie van c-commanderen af te leiden. Hij volgt Chomsky (1995) in de aanname dat syntactische structuur stukje bij beetje van onderen af wordt opgebouwd via het proces 'Merge'. Merge voegt twee elementen samen tot een constituent. De structuur in (3) ontstaat dus door eerst Z te combineren met *compl*, wat Z' oplevert, dan *spec* met Z', wat ZP oplevert, etc. De structuur groeit dus met elke stap. Epstein's idee is nu om de c-commandeerrelatie per stap te definiëren. Na de eerste stap hebben we de structuur in (4):



Z c-commandeert nu alle andere knopen in de boom (nl. *compl*). Na de tweede stap hebben we (5):



Nu c-commandeert *spec* alle knopen in de boom, nl. *Z'*, *Z*, en *compl*. Maar *Z* c-commandeert nog steeds alleen *compl*. De c-commandeer-eigenschappen van *Z* zijn eens en voor altijd vastgelegd op het moment dat *Z* gecombineerd werd met *compl*. Uit deze procedure volgt dat *Y* in (3) nooit *X* kan c-commanderen. Op het moment in de derivatie dat de c-commandeerrelaties voor *Y* bepaald worden *bestaat X nog niet eens*.

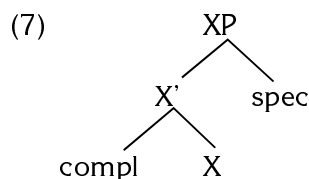
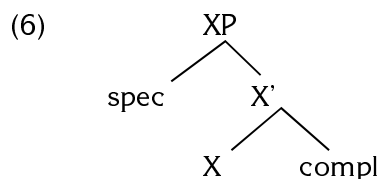
Volgens deze definitie van Epstein is c-commanderen dus een totale relatie tussen een element dat nieuw aan de structuur wordt toegevoegd en alle andere knopen in de structuur. Er hoeft dus op elk moment in de derivatie geen afbakening gemaakt te worden tussen elementen die wel en niet ge-c-commandeerd worden. Het probleem van de definitie van c-commanderen is daarmee opgelost.

Nu stelt Koster dat de eigenschappen van syntactische structuur fundamenteeler zijn dan de eigenschappen van de operatie Merge. Om de c-commandeerrelatie te kunnen definiëren moet Merge structuren als (3) opleveren, en de vraag waarom Merge dat doet is onbeantwoord.

Ik denk dat deze kritiek geen hout snijdt. Zoals ik hierboven al stelde bestaat over de aard van syntactische structuur geen controverse. (3) is voor de meeste doeleinden een geschikte manier om structuur af te beelden (maar Chomsky 1995:243 laat er geen misverstand over bestaan dat de *notatie* hier niet de kern van de zaak is). Tal van verschijnselen wijzen erop dat de syntaxis geen brei is maar een gelaagd, gestructureerd geheel van constituenten. Maar dat betekent niet dat *als* het resultaat van Merge een brei zou zijn (Koster's "Merge van Mars"), dat dan Epstein's definitie van c-commanderen niet meer zou opgaan.

Epstein definieert c-commanderen namelijk cruciaal niet aan de hand van het *resultaat* van de operatie Merge (wat we kunnen afbeelden als een structuur als in (3) of als een brei), maar aan de hand van het *moment* waarop de operatie Merge plaats vindt. Als element α gecombineerd wordt met een brei β , dan zal α β en alle elementen in β c-commanderen, ongeacht de structuur van de combinatie van α en β . Epstein's definitie is bij mijn weten de enige die niet cruciaal refereert aan de eigenschappen van de *notatie* van syntactische structuur (hoe geschikt die notatie ook is om de syntactische structuur te visualiseren).

Het gedachtenexperiment van "Merge van Mars" brengt ons op een ander punt. Stel dat syntactische structuren niet, zoals Kayne (1994) en Koster voorstellen, lineair geordend zijn als in (6), maar zo als in (7), of zelfs in het geheel niet lineair geordend:



In dat geval zou de klassieke definitie van c-commanderen voorspellen dat de c-commandeerrelaties ongewijzigd zouden blijven. De definitie van Koster zou echter tot gevolg hebben dat de c-commandeerrelaties omgedraaid zouden worden, of niet gedefinieerd zouden zijn. Deze situatie doet zich nooit voor, omdat we aangenomen hebben dat structuren er altijd uit zien als in (6). Dat maakt het ook mogelijk om de precedentie-relatie te laten meehelpen bij het afbakenen van het lokale domein voor de c-commandeerrelatie. Maar daarmee is de definitie van c-commanderen cruciaal afhankelijk van de toevallige omstandigheid dat structuren lineair geordend zijn als in (6). De klassieke definitie daarentegen is onafhankelijk van toevallige lineaire eigenschappen, en

Epstein's definitie heeft dat met de klassieke definitie gemeen. Epstein's definitie is dus principieel, waar die van Koster contingent is.

Nu kan men tegenwerpen dat precedentie een zo fundamenteel kenmerk is van de grammatica dat dat bezwaar niet telt. Niet voor niets neemt Koster precedentie op in zijn configurationele matrix: het antecedent moet altijd aan het afhankelijke element voorafgaan. Maar Chomsky (1995, § 4.8) relegeert de notie lineaire volgorde tot de fonetische component van de grammatica PF. Dat wil zeggen, lineaire relaties zijn volgens hem pas het resultaat van de omzetting van een abstracte syntactische structuur naar een ééndimensionale reeks klanken. Dat klanken lineair geordend moeten zijn (in de tijd) is onvermijdelijk, maar dat lineaire ordening een rol speelt in syntactische structuren is daarmee niet aangetoond.

Wat Koster precedentie noemt is dus mogelijk een fonetische pendant van een abstractere vorm van asymmetrie. Deze asymmetrie lijkt mij een fundamenteel onderdeel van de operatie Merge: het is een applicatie van één element aan een ander element (de 'target'). Daarmee voldoet Merge aan de configurationele matrix (de 'target' is het afhankelijke element δ , het toegevoegde element is het antecedent α , het lokale domein wordt door de operatie Merge zelf gegenereerd), zonder dat precedentie gestipuleerd hoeft te worden als een eigenschap van de configurationele matrix. De overige eigenschappen van de configurationele matrix (bi-uniciteit en bi-lokaliteit) kunnen niet afgeleid worden. Bi-uniciteit kan gezien worden als een primitieve die de vorm van de operatie Merge bepaalt. Bi-lokaliteit heeft in Koster (1998, 1999) als belangrijkste functie dat het de definitie van c-command afleidt, en is dus overbodig als de we definitie van Epstein aannemen.³

Ik zou dus willen suggereren dat de configurationele matrix minimaal, en idealiter maximaal, de eigenschappen in (8) heeft:

(8) Eigenschappen van de configurationele matrix

- (i) asymmetrie
- (ii) bi-uniciteit

De primitieve syntactische operatie Merge voldoet aan de configurationele matrix, en aan de hand van Merge kunnen we vervolgens c-commanderen definiëren (als een totale relatie tussen het 'gemerged' element en de (onderdelen van de) 'target').

Natuurlijk zouden we ook af kunnen zien van de klassieke clause in de definitie van c-commanderen die verwijst naar de eerste vertakkende knoop, en stellen dat elke knoop zijn zusterknoop c-commandeert en alles wat door zijn zusterknoop gedomineerd wordt. Maar vergeleken met de definitie van Epstein zou dat een stipulatieve definitie zijn die onmiddellijk de vraag oproept waarom de zuster-relatie relevant is voor de definitie van c-commanderen. Bovendien zijn relaties als 'zuster' en 'domineren' moeilijk te definiëren (buiten ostensie om), tenzij opnieuw gebruik wordt gemaakt van de primitieve operatie Merge.⁴

Samenvattend: de configurationele matrix bepaalt eigenschappen van de operatie Merge, maar de operatie Merge bepaalt de definitie van c-commanderen.

³ De definitie van bi-lokaliteit in Koster (1998) staat ook schendingen van de striktste definitie van c-commanderen toe, doordat het lokale domein waarbinnen bi-lokaliteit geldt variabel gedefinieerd kan worden. Ik ga er van uit dat de verschijnselen die daartoe aanleiding geven op een andere manier beschreven kunnen worden. Hetzelfde geldt voor de condities op gapping, besproken in Koster (1998:104).

⁴ Zuster: α is een zuster van β als α gemerged is met β . Domineren: α domineert β als α het resultaat is van een operatie Merge waar β bij betrokken was.

Referenties

Chomsky, Noam.

1995 *The Minimalist Program*. MIT Press, Cambridge.

Epstein, Samuel David, Erich M. Groat, Ruriko Kawashima, and Hisatsugu Kitahara

1998 *A Derivational Approach to Syntactic Relations*. Oxford University Press, New York.

Kayne, Richard S.

1994 *The Antisymmetry of Syntax*. MIT Press, Cambridge.

Koster, Jan.

1987 *Domains and Dynasties: the radical autonomy of syntax*. Foris, Dordrecht.

Koster, Jan.

1998 Gapping moet blijven. *TABU*28, 99-106.

Koster, Jan.

1999 De primaire structuur. *TABU*29, 131-140.