# Parseren in de polder
# Nederlandse taaltechnologie in perspectief

PETER-ARNO COPPEN EN CRIT CREMERS

**Abstract**

**This issue contains descriptions of four parsers of Dutch. Yet, building parsers is not a standard occupation among (Dutch) linguists. Computation deserves more attention in linguistics than it actually gets.**

## 1 Aanleiding

In januari 2001 organiseerde de Landelijke Onderzoeksschool Taalwetenschap (LOT) haar jaarlijkse winterschool, met workshops voor taalkundige AIO's. Een van die workshops, getiteld *Battle of the Parsers*, werd gegeven door de computerlinguïsten Gosse Bouma (RUG), Peter-Arno Coppen (KUN), Crit Cremers (UL) en Ton van der Wouden (UU). De workshop had tot doel de vergelijking van een viertal computerprogramma's voor de analyse van het Nederlands (*parsers*), en het vaststellen van de stand van zaken met betrekking tot het parseeronderzoek in de Lage Landen.

Deze workshop vormde de aanleiding tot dit themanummer. In dit nummer bespreken de vier docenten en leden van hun respectievelijke onderzoeksgroepen nog eens in wat uitgebreidere vorm hun parsers.

In deze inleiding op het themanummer bespreken de twee initiatiefnemers van de workshop in het kort de verhouding tussen taalkunde en technologie, de stand van zaken in het parseeronderzoek, en de perspectieven voor de toekomst.

## 2 Taalkunde en technologie

"Als menselijke gedragingen computerprogramma's waren, dan was menselijke taal de *killer application*", zo formuleerde de Amerikaanse journalist Franklin Cook in een artikel op de website <www.yourdictionary.com> het belang van menselijke taal. Het eigenaardige is dat deze stelling voor werkelijke computerprogramma's niet op lijkt te gaan. Computerprogramma's die menselijke taal verwerken worden alom aangekondigd als de technologie die ons leven in de komende decennia gaat beheersen (zie paragraaf 3 hieronder),

maar sceptici laten niet na op te merken dat er in de laatste dertig jaar maar weinig vooruitgang is geboekt in het automatisch verwerken van menselijke taal. Waar blijft de automatische vertaalmachine? Hoe zit het met de automatische grammaticacontrole? Waarom kunnen de duurste spellingcorrectieprogramma's op de grafsteen voor Pim Fortuyn het woord *nog* nog niet van *noch* onderscheiden?

Taalwetenschappers en technologen zijn het erover eens dat de verwerking van natuurlijke taal door computers geen louter technologische aangelegenheid is. Zelfs bij een haast zuiver natuurkundige exercitie als het herkennen van menselijke spraak is kennis van de taal die er gesproken wordt (het *taalmodel*) onontbeerlijk. En dat die kennis eigenlijk aangeleverd zou moeten worden door de taalwetenschap, ook daarover bestaat geen onenigheid.

Rond het midden van de vorige eeuw viel de opkomst van de moderne computer samen met een aantal opeenvolgende formaliseringstrends in de taalwetenschap, zoals structuralisme en generativisme. Het prachtige informatiekundige leerstuk van de Chomskyhiërarchie van talen, grammatica's en automaten getuigt nog daarvan. Al vrij snel in de confrontatie van taal en computer werd echter duidelijk dat echte alomvattende taalkundige analyse behoorde tot het soort taken waarvan niet gegarandeerd kan worden dat ze met beperkte inzet van middelen (geheugen, rekentijd, etc.) op computers uitgevoerd kunnen worden. Steekhoudende ontleding van natuurlijke taal behoort tot de ingewikkeldste computationele problemen die we kennen. De veelzijdige, ingebakken ambiguïteit van natuurlijke taal is de belangrijkste bron van deze complexiteit. Daarnaast bleek dat betrouwbare grammaticale modellen van natuurlijke taal al snel gigantisch groot en dus lastig te ontwikkelen en te onderhouden zijn. Dat vergt meer samenwerking, paradigmatische discipline en eensgezindheid dan taalkundigen lief lijkt. Toch danken grammaticamodellen als GPSG (Gazdar e.a. 1985) en HPSG (Pollard & Sag 1987) hun bestaan en hun ontwikkeling aan het computationeel gerichte onderzoek.

Voorts brak in technologische en andere niet-taalkundige kringen het inzicht door dat niet elke vorm van taalverwerking op strikt grammatische leest geschoeid hoeft te zijn. Veel interessante of bruikbare kenmerken van taal en tekst kunnen door statistische dataanalyse benaderd worden. Voor sommige kenmerken van teksten is het zelfs moeilijk andere dan statistische modellen voor de geest te halen.

Het gevolg van deze ontwikkelingen is dat hedendaagse taalverwerkende automaten voor een belangrijk deel weinig meer uitstaande hebben met de klassieke taalwetenschap. Zelflerende systemen worden afgestemd op grote hoeveelheden taalmateriaal, en de daarbij onstane taalmodellen hebben eerder het karakter van de neerslag van een statistische regelmaat dan van een grammaticaal regelsysteem. De meeste computerprogramma's die in een concrete taalverwerkende toepassing functioneren beperken zich dan ook tot een taalkundig oppervlakkige analyse: woorden en woordsoorten worden herkend en onderscheiden, en af en toe, afhankelijk van noodzaak in de beoogde toepassing, een stukje woordgroep (*chunk*).

Er is één domein waar technologen en taalwetenschappers elkaar op dit moment nog vinden: beiden erkennen het belang van de aanleg van grote taalcorpora ten behoeve van hun eigen onderzoek. Voor de technoloog is een taalcorpus een onontbeerlijke bron van trainingsmateriaal, en de taalwetenschapper vindt in het corpus de voedingsbodem voor linguïstische observaties of het materiaal waarop hypotheses empirisch getoetst kunnen

worden. Voor beide doeleinden is een rijk geannoteerd corpus het meest geschikt: een corpus waarin de woordvormen gerelateerd zijn aan lemma's, en benoemd naar hun categorie, liefst met verfijnde onderspecificaties (Oostdijk & Van Halteren 2001). Nog mooier wordt het als het corpus syntactisch geannoteerd is, dat wil zeggen: de uitingen voorzien van een syntactische structuur bestaande uit ten minste een woordgroepenverdeling en een benoeming van de woordgroepen en de relaties daartussen.

De vervaardiging van zo'n syntactisch geannoteerd corpus stelt technologen en taalkundigen echter voor een kip-en-ei-probleem: handmatige syntactische annotatie van een corpus met een beetje redelijke omvang (enkele miljoenen woorden) lijkt ondoenlijk. Dat zou dus automatisch moeten kunnen. Daarvoor is een parser nodig, een computerprogramma dat een syntactische analyse geeft van aangeboden taaluitingen. Gezien de combinatorische complexiteit waarmee zo'n parser bij de analyse van realistisch taalmateriaal geconfronteerd wordt, lijkt een statistische keuze van de goede analyse uit een veelheid van mogelijkheden noodzakelijk. Zo'n statistische keuze kan echter alleen gemaakt worden na training op grote hoeveelheden geannoteerd materiaal. Maar daarvoor is dan eigenlijk al de parser nodig die ermee getraind zou moeten worden.

Er zijn verschillende uitwegen mogelijk uit deze vicieuze cirkel: sommigen zoeken de oplossing in parsers die een minder diepe analyse nastreven (*shallow parsing*). Anderen zetten in op de (gedeeltelijke) automatisering van de keuze uit alle analyses (desambigueringssoftware of *discourse oriented parsing*). En ten slotte worden er steeds betere annotatiemiddelen ontwikkeld: software waarmee een in wezen handmatige analyse sneller gemaakt kan worden.

## 3 De huidige toestand

Wie taal door een computer wil laten verwerken, heeft vroeg of laat een *model* van die taal nodig, een beschrijving van de structurele kenmerken van die taal. Die zitten immers niet al in de bestaande processorchips ingebakken. In de loop van de geschiedenis van de taaltechnologie zijn er diverse soorten taalmodellen ontwikkeld. Voor taalverwerkende automaten bestaan op dit moment drie globale soorten architectuur:

- probabilistische modellen
- grammatische modellen
- subsymbolische modellen

Probabilistische modellen werken op basis van statistische analyses. Ze ontlenen de waarschijnlijkheden voor woordcombinaties en structuren of andere eigenschappen aan grote tekstbestanden (corpora). Op het eerste gezicht lijkt er in dergelijke modellen weinig taalkundige creativiteit geïnvesteerd; het gaat immers gewoon om tellen. Toch wordt de kwaliteit van zo'n probabilistisch model gemeten aan de kwaliteit van de analyse, en om die te garanderen moeten juist de relevante data aan het corpus zijn onttrokken. En het vergt wel degelijk taalkundig inzicht om te beslissen wat relevante data zijn.

Daarnaast is het inderdaad zo dat de in probabilistische modellen verwerkte inzichten niet per se deel hoeven uit te maken van een samenhangende taaltheorie; in de praktijk

blijkt er bij statistische analyse veel ruimte voor taalkundig opportunisme. Ook zonder een zingevend of totalitair taalkundig systeem kunnen talige verschijnselen op een intelligente manier met elkaar in verband worden gebracht. Aldus gebouwde modellen hebben daardoor vaak naast het nadeel van taalkundige ondoorzichtigheid het voordeel van robuustheid.

Grammatische modellen komen het dichtst in de buurt van de klassieke ontleder of generator. Op basis van een eindige hoeveelheid geformaliseerde taalbeschrijving worden zinnen herkend en gestructureerd of voortgebracht. Die formalisering is essentieel: een computer kan nu eenmaal niet gevoed worden met in vage regels of suggestief gestelde grammatica's. Grammatische modellen kunnen dan ook goed dienen om na te gaan hoe precies en nauwkeurig een bepaalde grammatica (theoretisch of descriptief) per saldo is. Men kan stellen dat dit eigenlijk de *core business* van de moderne taalkunde zou moeten zijn: formele toetsing van stelsels van beweringen over taal. Maar niet alle grammatica's lenen zich voor formalisering, en er zijn zelfs taalkundigen die volhouden dat grammatica's van natuurlijke taal nu eenmaal niet formaliseerbaar zijn.

Drie van de vier in dit nummer besproken ontleders horen tot deze klasse: Amazon, Alpino en Delilah. De onderliggende taalkundes lopen nogal uiteen, maar elk systeem heeft een expliciete grammatische ruggegraat. Amazon is gericht op *shallow parsing*, Alpino maakt gebruik van expliciete desambiguering, en Delilah combineert syntactische met semantische ontleding.

Subsymbolische modellen vormen de jongste ontwikkeling in de opzet van taalverwerkende automaten. Het model bestaat hier uit een netwerk van processoren die onderling met elkaar verbonden zijn en op elkaar reageren. Omdat de afzonderlijke taken van iedere processor niet corresponderen met een klassieke taalkundige of technologische taak (een symbool), spreken we van subsymbolische processen. Een dergelijk netwerk als geheel voert dan bepaalde –tot op heden: elementaire– taalkundige taken uit, zoals het toekennen van woordklassen aan de woorden in een tekst, of het herkennen van constituenten. Daartoe wordt het "getraind", dat wil zeggen het wordt gevoed met invoer en beoogde uitvoer, net zolang totdat het zelf heeft "ontdekt" hoe de beoogde uitvoer het beste kan worden afgeleid van de invoer.

Het bijzondere van zo'n training is dat er geen expliciete  informatie over de aard van de taak aan het netwerk wordt aangeboden. Als een netwerk erin slaagt na training een taak uit te voeren, is de status van het netwerk op dat moment het model van het taakdomein. Het netwerk is zo de grammatica. De CGN-parser draait op een subsymbolisch tot stand gebrachte annotatie en bedrijft *shallow parsing* ten behoeve van een corpus.

Het ligt voor de hand dat er ook allerlei hybride vormen van taalkundige modellen in zwang zijn. Elk van de modeltypes heeft z'n eigen merites. De probabilistische modellen lenen zich het best voor beschrijving en analyse van *taal-in-extensie*: herkenning en analyse van concreet taalgebruik en productie van genormaliseerde taal. De grammatische modellen implementeren bij definitie *taal-in-intensie*, zoals het taalvermogen, voor zover te onderscheiden als een apart cognitief systeem. De subsymbolische systemen leveren weinig informatie over taalstructuur, omdat het model niet expliciet is, maar zijn juist buitengewoon geschikt om leer- en verwervingsprocessen te analyseren en te simuleren.

In Nederland wordt het modelleren van taal nauwelijks opgevat als een deel van het taalkundige ambacht. Computationele taalkunde is in verreweg de meeste linguïstische

programma's aan de universiteiten niet vertegenwoordigd. De instellingen bieden doorgaans alleen bijvakprogramma's aan –vaak verzorgd door aparte opleidingen als alfa-informatica. Het is niet overdreven om te stellen dat er beduidend meer computationele taalkunde buiten dan binnen het taalkunde-onderwijs te vinden is: bij informatici, psychologen en logici bijvoorbeeld, en in aparte programma's voor alfa-informatica.

In het onderzoek is de situatie niet veel anders. Een minderheid van computationeel actieve taalkundigen is verbonden aan de taalkunde-opleidingen. Alle auteurs in dit nummer zijn oorspronkelijk neerlandici en bouwen ontleders van het Nederlands, maar geen werkt bij een opleiding Nederlandse taalkunde. Overal in de wereld wordt gewerkt aan allerlei taaltechnologische toepassingen, en de taalkundigen lopen meestal niet voorop. Ook hier lijkt sprake van een duivelse cirkel: omdat veel bestaande talige toepassingen van informatietechnologie slechts in bescheiden mate taalkundig zijn gefundeerd, beschouwen taalkundigen dit domein als beunhazerij, zonder dat ze zelf bereid of in staat zijn een betere bijdrage te leveren. Bijgevolg laten ze de kansen over aan taalkundig minder bezwaarden, die dan weer succesvolle systemen ontwikkelen zonder al te veel taalkunde.

Overal waar informatietechnologie opdringt, dienen zich perspectieven aan voor taalkundig werk. In Nederland is een tak van het Openbaar Vervoer Informatie Systeem ingericht met behulp van een natuurlijke taal-*interface*: de mogelijkheid om met behulp van gesproken taal een computer gesproken informatie te ontlokken over het treinverkeer. Voor dit systeem zijn naast elkaar twee taalkundige modellen ingezet: één op statistische en een op grammatische grondslag, geleverd door de afdelingen alfa-informatica van respectievelijk de UvA en de RUG. De beide systemen zijn uitvoerig geëvalueerd (Veldhuyzen van Zanten e.a., 1999). Het OVIS project heeft relatief veel impact gehad, al lijkt het niet op de uitwerking van het Duitse *Verbmobil*-project (<http://verbmobil.dfki.de/overview-us.html>) dat vijftig miljoen euro en navenant veel taalkundige onderzoekers op vernieuwende wijze van de straat heeft gehouden. Ook in allerlei Nederlandse bedrijven wordt gewerkt aan gerichte toepassingen, variërend van robuuste herkenning van namen tot aan vertaalsystemen.

Sceptici terzake kunnen wijzen op de deconfiture van de Vlaamse taal- en spraaktechnologiegigant Lernout en Hauspie en op het einde van het Eindhovense Rosetta-project. In beide gevallen is deze teloorgang echter niet te wijten aan technologische – laat staan aan taalkundige – maar aan bedrijfskundige tekortkomingen. In het geval van Rosetta, een ambitieus, op taalkundige inzichten gebaseerd automatisch vertaalsysteem, is weliswaar het gebrek aan baten op korte termijn aangevoerd om het project af te bouwen maar de taalwetenschappelijke en vertaaltechnische opbrengst van het project is meer dan aanzienlijk (Rosetta 1994).

Hier zit overigens wel een probleem: taalkundig verantwoorde toepassingen zijn vaak zo arbeidsintensief en complex dat de ontwikkeling ervan altijd economisch kwetsbaar zal zijn. Niet alle vraag naar taaltechnologie drijft op het bewustzijn dat een kwalitatief goed product geld en tijd kost. Niettemin wordt er veel meer bedacht en al dan niet succesvol geknutseld dan een rondgang langs de huisaltaren van de vaderlandse academische taalkunde zou doen vermoeden. Tegelijk zijn er weinig, te weinig taalkundigen bij betrokken. Dat is slecht voor de toepassingen, en buitengewoon belabberd voor de taalkunde.

## 4 Perspectieven

Ook niet-profeten kunnen voorzien dat de confrontatie tussen taal en computer een van de wedstrijden van de eeuw wordt. De informatie die wij over elkaar uitstorten, neemt in de tijd exponentieel toe. Alle informatie is vroeg of laat talig: de waarheid is een zin, geen grafiek en ook geen plaatje. De mogelijkheden voor mensen om die informatie te absorberen en te verwerken nemen nauwelijks toe. Als we dan die informatie wel serieus nemen en geïnformeerdheid beschouwen als een voorwaarde voor meepraten, is het in ieders belang om de instrumenten voor opslag, ontsluiting en weergave van informatie drastisch te verbeteren.

De afgelopen decennia zijn er allerlei technieken en technologieën ontwikkeld die de informatisering schragen, zoals de *markup*talen (html, bijvoorbeeld) en databases. Deze technologieën organiseren en reorganiseren informatie, maar parasiteren hierbij op de vorm van informatie en veronderstellen opgelegde orde. Noodzakelijkerwijs blijven deze instrumenten ver van de inhoud, en zeker ver van de interpretatie van (talige) informatie. Er is hier een wereld te winnen voor diepstekende, subtiele en ambitieuze taalkunde: het ontwikkelen van instrumenten die op informatie-inhoud kunnen opereren.

Wat minder ambitieus lijken al die taalhulpen die de tekstverwerkers zeggen te hebben maar die het vooralsnog niet zo goed doen. Het probleem is duidelijk: zelfs een eenvoudige spellingchecker vergt bijna totale syntacto-morfologische analyse, en dat kost geld. Maar niemand stapt van Word naar WP over omdat die beter *d*'s van *t*'s onderscheidt, wanneer de software daardoor drie keer zo duur wordt. Er valt moeilijk te concurreren met betere taalhulpmiddelen. Betrouwbare en interessante spelling- en grammaticacheckerij vallen dus samen, en zullen eerder als *spin-off* van grotere ondernemingen in de tekstverwerking terechtkomen dan als zelfstandig ontwikkelingsdoel.

Zelfs profeten kunnen niet voorspellen wat de aard van de inhoudsgerichte instrumenten zal zijn die uiteindelijk de informatie-explosie moeten kanaliseren. Het kan best zijn dat regelgestuurde taalkunde niet of niet goed opgewassen is tegen eisen van robuustheid, onderhoudbaarheid of economie. Anderzijds is het zeer wel denkbaar dat de statistische modellen stuitend onprecies blijken als het om betekenis en duiding gaat, of dat de subsymbolische modellen voor inhoudelijke analyse niet in menselijke tijd trainbaar zijn.

Misschien zullen er nooit ergens voldoende brein en pecunia te vinden zijn om veelomvattende taaltechnologische systemen te ontwikkelen. Maar wellicht ook zal ooit –om een gedachte van de Utrechtse taaltechnoloog Steven Krauwer te volgen– een publiek-private *joint-venture* een man-naar-de-maan-project voor de computationele taalkunde entameren, waarin taalkunde een *core*-technologie wordt met uitstraling en doorwerking op vele aanverwante terreinen.

Het meest waarschijnlijke is echter dat er in de nabije toekomst door kleine groepen tot op heden niet vertoonde mengvormen van benaderingen en subsystemen worden geconstrueerd, zoals dat ook bij het eerder genoemde Duitse *on-line*-vertaalsysteem *Verbmobil* is gebeurd. Taalkundigen zouden het voortouw moeten nemen bij de ontwikkeling, toetsing en inzet van dergelijke hybride taalmodellen.

In ieder geval zal de taalkunde zelf er niet aan ontkomen de grenzen van de berekenbaarheid van taal op te zoeken: welke aspecten van natuurlijke taal laten zich formeel modelleren en welke niet of niet goed? Het is duidelijk dat de gemiddelde zinslengte in

geschreven Nederlands bepaalbaar is, en het is anderzijds ook duidelijk dat psychologische en sociale connotaties van de zin *ik zeg wat ik denk* nooit geheel berekenbaar zullen zijn. Maar daartussen liggen nog talloze andere eigenschappen van taal besloten, en ergens gaat het berekenbare in het niet- of niet-goed-berekenbare over. Wanhopig maar vastberaden zoeken van die berekenbaarheidsovergangen is een hoge vorm van taalwetenschap, en bijna een *mission statement* voor de linguïstiek.

## Bibliografie

**Bouma, G. en I. Schuurman (2000).** De digitale infrastructuur van het Nederlands. *Nederlandse Taalkunde* 5, 90-94.

**Gazdar, G., E. Klein, G. Pullum en I. Sag (1985).** *Generalized Phrase Structure Grammar*. Oxford: Blackwell.

**Oostdijk, N., en H. van Halteren (2002).** De grammaticale annotatie van tekstcorpora. *Nederlandse Taalkunde* 7, 175-181.

**Pollard, Carl en Sag, Ivan (1987).** *Information-Based Syntax and Semantics: Vol. 1 – Fundamentals*. Stanford, California: CSLI.

**Rosetta, M.T. (F.M.G. de Jong, T.M.V. Janssen, L. Appelo & J. Landsbergen, eds.) (1994).** *Compositional Translation.* Dordrecht: Kluwer.

**Veldhuyzen van Zanten, G., G. Bouma *e.a.* (1999).** Evaluation of the NLP Components of the OVIS2 Spoken Dialogue System. <http://grid.let.rug.nl:4321/public-docs/art84.ps>

# Het geheim van de oude dame
# De Nijmeegse parser Amazon

PETER-ARNO COPPEN*

**Abstract**

**Originated in ancient times, the Amazon parser for Dutch appears to be a worthy competitor among contemporary rivalling parsers. In this article the causes of this success are discussed. From a concise history, Amazon's main characteristics are derived: it is a shallow parser, based on a structuralist descriptional theory. Moreover, Amazon's aims are modest: the Amazon parse is meant as only a first step in a total analysis. Subsequent components are needed to refine the Amazon parse tree.**
**Three main trends are discussed that characterize the Amazon development: modularization, separation of linguistic theory from algorithm, and the development of robustness strategies, which have led to the current Amazon parser. Next, Amazon's performance is briefly evaluated. In conclusion, it is argued that shallow parsing is a suitable first step in parsing natural language. Shallow parsing can even be motivated from linguistic theory.**

## 1 Vooraf

Parsers hebben geen geschiedenis. Ze ontstaan in de geest van de tijd, worden –met een beetje geluk– toegepast en raken ingehaald door nieuwe ontwikkelingen. Er is voor een parser blijkbaar geen wezenlijke verdienste gelegen in een lange levensduur. Oud is ouderwets, nieuw is het magische woord.

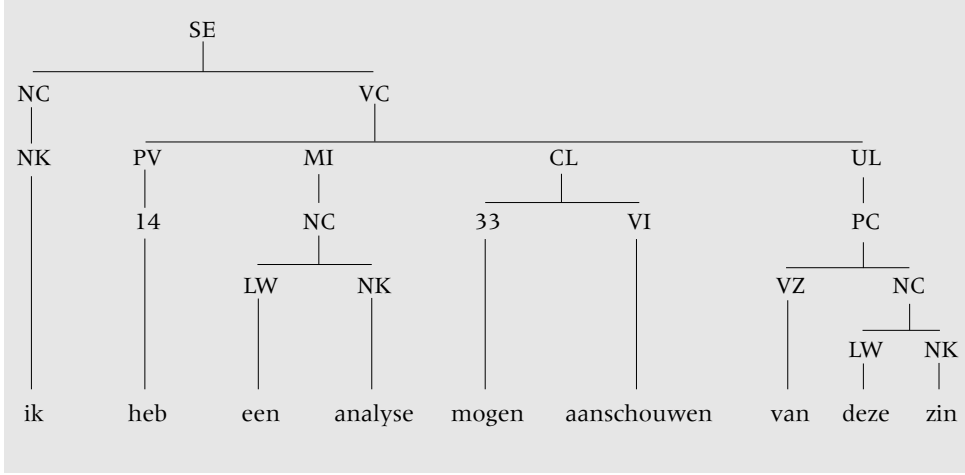
De Nijmeegse parser Amazon is in dit opzicht een buitenbeentje op het slagveld van de Nederlandse parsers. Haar geschiedenis gaat terug tot 1975, wat voor de technologie nog zo ongeveer het Stenen Tijdperk betekende. Ponskaarten, magneetbanden en computerprogramma's die 's avonds bij de operator moesten worden ingeleverd om 's nachts te worden uitgevoerd. Het verbeteren van een typfout kon zo een hele werkdag in beslag nemen.

Ondanks deze hoge ouderdom is de Amazonparser heden ten dage nog springlevend. Dat is te danken aan haar bescheiden doelstellingen, en een aantal verjongingskuren die

* Vakgroep Taal en Spraak, KU Nijmegen. Ik dank Carla Schelfhout, Jan Smeets, Bram Elffers en de reviewers van Nederlandse Taalkunde voor commentaar op een vorige versie van dit artikel.

zij in de loop der jaren heeft ondergaan. Toch is Amazon nog steeds Amazon, een oppervlakteparser[1] voor Nederlandse zinnen, op structuralistische grondslag.[2] Het lijkt zelfs meer dan ooit duidelijk dat de Amazonmethode, oppervlakteparsering, een aantrekkelijke strategie is als een eerste fase op weg naar een volledige analyse van de zin. Dat is de reden dat Amazon zich in een "Battle of the parsers" nog steeds durft te meten met haar jongere zusters.

De Amazonparser kan het best gekarakteriseerd worden aan de hand van haar historische ontwikkeling, in tandem met die van haar vervolgmodule, Casus geheten, die de verrijking van de Amazonparsering met thematische functies beoogt. In dit artikel zal ik deze geschiedenis daarom in het kort nalopen. En natuurlijk ga ik daarbij op zoek naar het Geheim van de Oude Dame: wat zijn de sterke punten van de huidige Amazonparser? Hoe zijn haar prestaties te verklaren tijdens de "Battle of the Parsers" op de winterschool 2001 van de Landelijke Onderzoeksschool Taalwetenschap (LOT)?




*Figuur 1: Analyse Amazon 1975*

## 2 De prehistorie

Eigenlijk begint de voorgeschiedenis van de Amazonparser al in 1928, toen de taalkundigen Rijpma en Schuringa hun Nederlandse Spraakkunst schreven. Deze op structuralistische leest geschoeide schoolgrammatica werd in de daaropvolgende jaren erg populair. Na 21 drukken nam de Nijmeegse taalkundige Jan van Bakel het in de tweede helft van de jaren zestig op zich om de 22e druk te moderniseren.

1 De internationale term is *shallow parser*. Het betreft natuurlijk geen oppervlakkige parser in de pejoratieve zin. De term verwijst naar de structuren die de parsering oplevert, en die als oppervlaktestructuren kunnen worden gekarakteriseerd. De hedendaagse term *shallow parser* bestond in 1975 uiteraard nog niet.

2 Amazon is gebaseerd op de structuralistische grammatica van Rijpma & Schuringa (1968). In de loop van de tijden zijn verschillende delen van de grammatica echter aangepast aan modernere inzichten. Niettemin is Amazon nog steeds een structuralistische grammatica te noemen.

Bij deze herwerking werd hij getroffen door de sterke formalisatie die het werk kenmerkte. Aangezien Van Bakel al eerder experimenten met de analyse van gedigitaliseerde teksten had uitgevoerd, vroeg hij zich af of de reikwijdte van de Rijpma & Schuringa-formalisatie ook meetbaar was met behulp van een computerprogramma. Hoe goed wás de Rijpma & Schuringa-grammatica eigenlijk? Bood zij een analysemodel voor alle Nederlandse zinnen?

Om deze onderzoeksvraag te toetsen, implementeerde Van Bakel het structuralistische formalisme van Rijpma & Schuringa in een computerprogramma dat hij Amazon noemde (een acroniem voor **A**uto**MA**tische **Z**ins**ON**tleding). Dit computerprogramma nam een Nederlandse zin (of een deel daaruit) als invoer, en gaf van deze zin een analyse volgens Rijpma & Schuringa.

De eerste Amazonparser was dus een computerprogramma. Het was *interactief*, dat wil zeggen het kon via een terminalverbinding met de mainframe computer direct worden uitgevoerd in een dialoog met de gebruiker. Daarnaast was het *syntax embedded*: de taalkundige theorie was niet in een apart formalisme gescheiden van het algoritmische deel van het programma, maar versleuteld in het programma zelf. Dat programma was weliswaar geschreven in een van de hogere programmeertalen uit die tijd (het in tekstverwerking gespecialiseerde SNOBOL4[3]) maar daarmee konden geen taalkundige regels worden weergegeven in de vorm waarin dat destijds gebruikelijk was (herschrijfregels en transformatieregels).

Hoewel Amazon dus geen expliciete formalisatie van taalkundige regels bevatte, lag zo'n formalisatie wel ten grondslag aan het parseeralgoritme. Het Amazonprogramma is achteraf te reconstrueren als een topdown links-rechts parsering,[4] met enkele handige optimalisaties waarvan de meeste uit nood geboren waren vanwege de beperkingen van de toenmalige hardware.[5] Deze parsering volgde een tamelijk rechttoe rechtaan herschrij-

**3** SNOBOL4 staat voor StriNg Oriented and symBOlic Language. Dit anarchistische acroniem dreef de spot met het verschijnsel dat voor elke naam van een computerprogramma vaak achteraf een acroniem in elkaar werd geknutseld. De naam Amazon neemt deze vorm van satire over. Overigens draaide Amazon onder een dialect van SNOBOL4, het zogeheten SPITBOL (Speedy Implementation of snoBOL4), dat de acroniemensatire nog een stapje verder voerde. De programmeertaal SNOBOL4 werd door Van Bakel als meest geschikt voor parsering beoordeeld, omdat zij beschikte over uitgebreide faciliteiten voor patroonherkenning, het herkennen van patronen in teksten. Tegenwoordig geldt SNOBOL4 als een ouderwetse taal, maar de SNOBOL-patroonherkenning wordt alom geprezen en nagevolgd. De hulpprogramma's van het huidige Amazonsysteem zijn nog steeds in SPITBOL geschreven.

**4** Een algoritme waarbij de analyse uitgaat van het beginsymbool van de grammatica, en de woorden in de zin van links naar rechts worden afgewerkt. De tegenhanger van het top down parseren is het bottom up parseren, waarbij de parser uitgaat van de woorden in de zin en van daaruit een analyse probeert op te bouwen. Tegenwoordig bestaan er verschillende mengvormen.

**5** De analysemethode is gerelateerd aan de prestaties van de parser. Ruwweg gezegd: top down parseren kost tijd, en bottom up kost geheugenruimte. Aangezien dat laatste een meer praktische beperking was (de machines hadden toentertijd een beperkte geheugenruimte), was top down parsering vaak de voor de hand liggende keuze. De optimalisaties die Van Bakel in Amazon doorvoerde waren gericht op het beperken van het zogeheten 'backtrack-effect': doordat de parser bij verkeerde keuzes in principe op alle eerder genomen beslissingen moet kunnen terugkomen, ontstaat snel een explosie van mogelijkheden die moeten worden onderzocht. Van Bakel beperkte onder andere eenvoudigweg de diepte van de mogelijke parseerboom tot een interactief vast te stellen waarde.

ving, die de structuralistische velden en constituenten in de zin markeerde. Figuur 1 geeft een indruk van zo'n beschrijving.[6]

Deze voorbeeldanalyse laat duidelijk de mengeling zien van structuralistische velden en constituenten. De knopen MI en UL zijn structuralistische velden (resp. het middenveld en de uitloop), terwijl knopen als VC, NC en PC typische constituenten zijn. In latere versies van Amazon is deze basisstructuur regelmatig bijgesteld maar de oude indeling is nog steeds herkenbaar.

Tegenwoordig verdeelt Amazon elke Nederlandse zin in zeven structurele velden: het middenveld als kern, omgeven door twee werkwoordelijke polen: de werkwoordelijke eindgroep rechts van het middenveld, en de persoonsvorm of het voegwoord links. Vóór de persoonsvorm ligt het topicalisatieveld, nog voorafgegaan door het links-dislocatieveld (of: aanloop). Rechts van de werkwoordelijke groep ligt het extrapositieveld, nog gevolgd door het rechts-dislocatieveld (ook *uitloop* of *after thought* genoemd). In het volgende voorbeeld zijn alle velden gevuld:

(1) Zeg Jan, | tegen wie | heb | jij toch al die verhalen | verteld | over mij, | met die onsmakelijke details?

De buitenste twee velden zijn duidelijk door komma-intonatie gescheiden van de andere velden. De binnenste vijf velden vormen de vijf velden van de kernzin, die in de oudste versie van Amazon al aanwezig waren. In recente onderzoeksprojecten (Van Dreumel 1997, Gerrits 2001) worden binnen het middenveld nog nadere onderscheidingen gemaakt: zo wordt de clitische groep en de partikelgroep aan het begin apart gemarkeerd, en de afsluitende groep aan het eind met resultaatbepaling, richtingbepaling, idioom, predikaat of r-partikel.

Een opvallende eigenschap van de Amazonanalyse is het ontbreken van vrijwel elke semantische informatie. Natuurlijk is de syntactische structurering zelf in zekere zin de uitdrukking van een vorm van semantische informatie (namelijk de informatie welke woorden bij elkaar horen), maar het feit bijvoorbeeld dat *we* het onderwerp van de zin is en *een analyse van deze zin* het –discontinue– lijdend voorwerp, is in de boomstructuur niet terug te vinden.[7] Deze beperking hangt samen met de structuralistische uitgangspunten van Amazon. De thematische informatie werd geacht onder de semantiek van de zin te vallen, en niet onder de structuralistische syntaxis.

Een onderdeel van het programma dat een aparte algoritmiek had, was de routine die de werkwoordelijke groep analyseerde. Deze routine was gebouwd op de vormverwachtingen van het werkwoord. Kort gezegd komt dit hierop neer: op elk moment tijdens de parsering staat er een vormverwachting uit ten aanzien van het volgende werkwoord. Dat kan een persoonsvorm zijn, een infinitief met of zonder *te*, of een voltooid of tegenwoordig deelwoord. Bepaalde werkwoorden of constructies beïnvloeden die verwachting. Een hulpwerkwoord van tijd zet de verwachting voor een voltooid deelwoord aan, een

**6** Deze en de volgende structuren zijn vereenvoudigd. Omwille van de helderheid van presentatie zijn de featurestructuren bij de knopen weggelaten.

**7** Eigenlijk is dit *thematische* informatie. De vraag of thematische informatie semantisch is, laat ik hier buiten beschouwing. Dat werd in 1975 in elk geval wel zo gezien.

beknopte bijzin met *om* of *teneinde* verwacht een infinitief met *te*. Een werkwoord wordt alleen geaccepteerd als het beantwoordt aan de uitstaande verwachting. Als alle verwachtingen zijn ingelost, is het einde van de werkwoordelijke groep bereikt.

Een voorbeeld van een werkwoordelijke groep die aan dit basispatroon beantwoordt is:

(2) We *schijnen* het werkstuk zaterdag *te moeten hebben ingeleverd*.

De zin begint met een initiële verwachting *persoonsvorm*. Het werkwoord *schijnen* lost die verwachting in, maar creëert zelf de verwachting van een infinitief met *te*. Die wordt weer ingelost door *te moeten*, dat op zijn beurt de verwachting van een infinitief zonder *te* activeert. Het hulpwerkwoord van tijd *hebben* is mogelijk zo'n infinitief, maar die roept weer de verwachting van een voltooid deelwoord op. Dat voltooid deelwoord is *ingeleverd*. Dat is een zelfstandig werkwoord zonder eigen verwachting, zodat het de werkwoordelijke groep afsluit.

Uiteraard zijn veel werkwoordvormen ambigu (zo kan *hebben* ook een persoonsvorm zijn met een verwachting "infinitief met *te*", als in *ze hebben dat maar te doen*) maar Amazon zoekt naar de optimale combinatie van lexicale mogelijkheden.

Speciale werkwoordelijke constructies worden in Amazon beschreven als afwijkingen van dit basispatroon. Zo verantwoordt de oorspronkelijke Amazon al de afwijkende plaats van het voltooid deelwoord (*we schijnen het werkstuk zaterdag ingeleverd te moeten hebben*), het werkwoordpartikel (*we schijnen het werkstuk zaterdag in te moeten hebben geleverd*) en de IPP-constructie[8] (*we schijnen het werkstuk zaterdag te hebben moeten inleveren*). In latere versies is deze beschrijving nog nader verfijnd en uitgebreid (zie voor een uitgebreide bespreking Van Dreumel & Coppen (te versch.)).

In figuur 2 is het transcript weergegeven van een interactieve sessie met de allereerste versie van Amazon (nu nog beschikbaar als SNOBOL4-programma onder de naam Amazon75). Te zien is dat de gebruiker het parseerproces kan sturen door de diepte van de analyse aan te passen, en de keuze voor lexicale items te beïnvloeden. Dat lijkt op een zwaktebod van de parser, maar in het oog moet worden gehouden dat de doelstelling van het Amazon-programma slechts een "proof of principle" was: de onderzoeksvraag was of het beschrijvingssysteem Rijpma & Schuringa *in principe* geschikt was voor elke Nederlandse zin. De parseerefficiëntie diende alleen dit praktische doel, de computer was slechts een hulpinstrument bij de beantwoording van een vraag die in theorie ook handmatig kon worden beantwoord.

Ook de taggingcomponent[9] van Amazon is in dit transcript af te lezen: Amazon heeft voor de structureel dubbelzinnige woorden (*heb*, *mogen*, *deze*) de correcte keuzes gemaakt.[10]

**8** In de Infinitivus Pro Participio-constructie (IPP) staat een infinitief op de plaats waar een voltooid deelwoord verwacht wordt. In het voorbeeld vraagt het hulpwerkwoord *hebben* eigenlijk om een voltooid deelwoord *gemoeten*.

**9** Onder *tagging* verstaan we het toekennen van een woordklasse aan elementen in de taaluiting. Zie ook Oostdijk & Van Halteren (2002). Een computerprogramma dat tagging verricht heet een *tagger*.

**10** Respectievelijk hulpwerkwoord van tijd (HVTP), hulpwerkwoord in de vorm van een infinitief met de verwachting infinitief (HVII), en attributief gebruikt demonstrativum (ILLE).

## 3 Uitbouw en afbraak

```
* * * * *      ZITTING  AMAZON        * * * * *
HET IS VANDAAG 02/18/02 14:20:47
ATTENTIE: HET DIEPTEBEREIK IS: 8
?
 DEBUG KLAAR OP VERZOEK.
dpt(5)
MAXIMALE DIEPTE GESTELD OP 5
ads(analyse)
—ANALYSE— TOEGEVOEGD AAN DE GRAMMATICA
ads(zin)
—ZIN— TOEGEVOEGD AAN DE GRAMMATICA
!
EINDE DEBUG—
HOE VERDER? — ANTWOORD “RETURN”, “START” OF “END”
start
ik heb een analyse mogen aanschouwen van deze zin.
1E ANALYSE; START DOOR MET ENTER OF “DEBUG”
** MET SUCCES ONTLEED IN 550 MSEC. **
TYP “JA” VOOR ANALYSE VAN HET RESULTAAT
NK  IK
NC  IK
14  HEB
LW  EEN
NK  ANALYSE
NC  EEN ANALYSE
MI  EEN ANALYSE
33  MOGEN
VI  AANSCHOUWEN
CL  MOGEN AANSCHOUWEN
VZ  VAN
LW  DEZE
NK  ZIN
NC  DEZE ZIN
PC  VAN DEZE ZIN
UL  VAN DEZE ZIN
VC  HEB EEN ANALYSE MOGEN AANSCHOUWEN VAN DEZE ZIN
SE  IK HEB EEN ANALYSE MOGEN AANSCHOUWEN VAN DEZE ZIN
?
 DEBUG KLAAR OP VERZOEK.
dmp(words)
WORDS<1> = IK
WORDS<2> = HVTP
WORDS<3> = EEN
WORDS<4> = ANALYSE
WORDS<5> = HVII
WORDS<6> = AANSCHOUWEN
WORDS<7> = VAN
WORDS<8> = ILLE
WORDS<9> = ZIN
!
EINDE DEBUG—
HOE VERDER? — ANTWOORD “RETURN”, “START” OF “END”
end
EINDE ZITTING AMAZON
HET AANTAL AANGEBODEN GROEPEN WAS: 1
```

*Figuur 2: Een sessie in  Amazon 75*

Zo rond 1980 werd duidelijk dat de oorspronkelijke onderzoeksvraag van het Amazonproject positief kon worden beantwoord. In vijf jaar was Van Bakel geen zinnen tegengekomen die niet in principe door het Amazonprogramma konden worden geanalyseerd en die daar naar zijn oordeel wel voor in aanmerking zouden moeten komen.[11] Amazon gaf in sommige gevallen ook wel andere, en onjuiste analyses, maar steeds kon de parser het bewijs leveren dat het beschrijvingssysteem van Rijpma & Schuringa in de computerimplementatie van Amazon de aangeboden zin in elk geval ook correct afdekte. Daarmee was het Amazonproject geslaagd, maar was de parser plotseling beroofd van zijn onderzoeksvraag.

Wat te doen met een parser die geen duidelijk doel meer heeft? Weggooien was zonde, en de parser werd dus hergebruikt in een nieuwe onderzoeksvraag: is het mogelijk om Nederlandse zinnen te analyseren tot een dependentiestructuur geïnspireerd op de casustheorie van Fillmore (Fillmore 1968)? Dit is een wezenlijk andere vraag dan de Amazonvraag. Immers, Amazon streefde "slechts" een structuralistische analyse na, waarin constituenten in hun velden werden benoemd maar niet voorzien van thematische informatie. In een Amazonanalyse kon je wel zien dat een constituent een NC was aan het begin van Middenveld, maar niet of die NC het onderwerp van de zin was.

Waarom koos Van Bakel voor de casustheorie van Fillmore, en niet voor de traditionele ontleding in onderwerp of lijdend voorwerp? Dat lag hieraan, dat de onderzoeksvraag meer als een semantische vraag dan als een syntactische vraag werd gezien. Het doel was niet langer de toetsing van een syntactisch beschrijvingssysteem, maar eerder informatietechnologisch van karakter: kunnen de betekenisverhoudingen in de zin worden opgespoord in een automatische analyse? Toch bleef die onderzoeksvraag in de praktijk taaltheoretisch van aard: ook in 1980 waren werkelijke toepassingen nog ver weg.

Ter beantwoording van die nieuwe vraag werd een tweede module ontwikkeld,[12] nu simpelweg *Casus* geheten (geen acroniem dit keer). In die tijd schrok men ervoor terug om het Amazonprogramma nader te compliceren, om verschillende redenen:

- Het programma was naar toenmalige maatstaven al erg groot en complex geworden, hetgeen een substantiële uitbreiding in de weg stond. Niet alleen voor de programmeur, maar ook voor de toenmalige machines, zou het programma al snel teveel worden;
- De uitbreiding was wezenlijk anders van karakter dan de Amazonparsering: was de laatste een klassieke topdown parsering tot een constituentenstructuur van de oorspronkelijke woordvolgorde, de Casus-uitbreiding zou een transformatie moeten inhouden naar een totaal andere structuur –de dependentiestructuur– waarin de woorden in een andere volgorde zouden komen te staan dan in de oorspronkelijke zin.

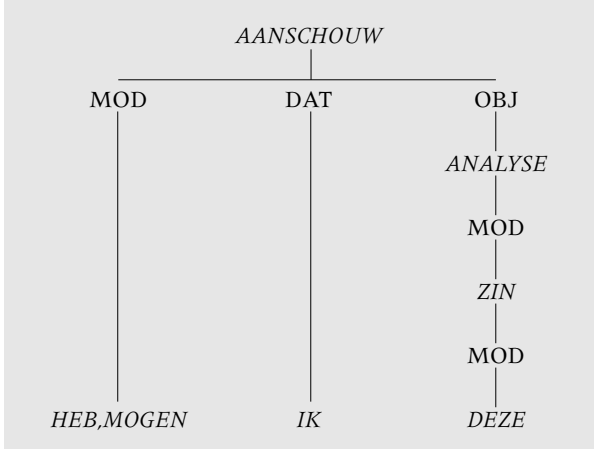
**11** Er bestond in die tijd nog levendige discussie over grammaticale en ongrammaticale zinnen. De laatste werden niet geacht onder de definities van een spraakkunst te vallen. Later, in de jaren negentig, verschoof deze opvatting naar een grotere nadruk op de *robuustheid* van de parser, dat wil zeggen: ook afwijkende en zelfs ongrammaticale invoer moest geanalyseerd kunnen worden.

**12** Ook weer een syntax embedded SNOBOL4-programma.

Samen met zijn doctoraalstudenten Computerlinguïstiek schreef Van Bakel in 1980 het programma Casus, dat de Amazonanalyse als invoer nam en op basis van een apart gedefinieerde set van volgorderegels voor Casusrollen[13] één of meer dependentiestructuren opleverde. In figuur 3 zien we een voorbeeld van zo'n Casusanalyse. De casusrollen hebben voor de hand liggende afkortingen.[14] De werkwoorden zijn de semantische kernen, en de casusrollen en eventuele modificeerders zijn hun dependenten. De hulpwerkwoorden en determiners worden geanalyseerd als kenmerken van het werkwoord of het zelfstandig naamwoord (in de figuur zijn deze kenmerken weggelaten).

In latere versies van Casus is het idee van de dependentiestructuur verlaten, en wordt de oppervlaktestructuur alleen verrijkt met thematische informatie. Figuur 4 geeft daarvan een indruk.

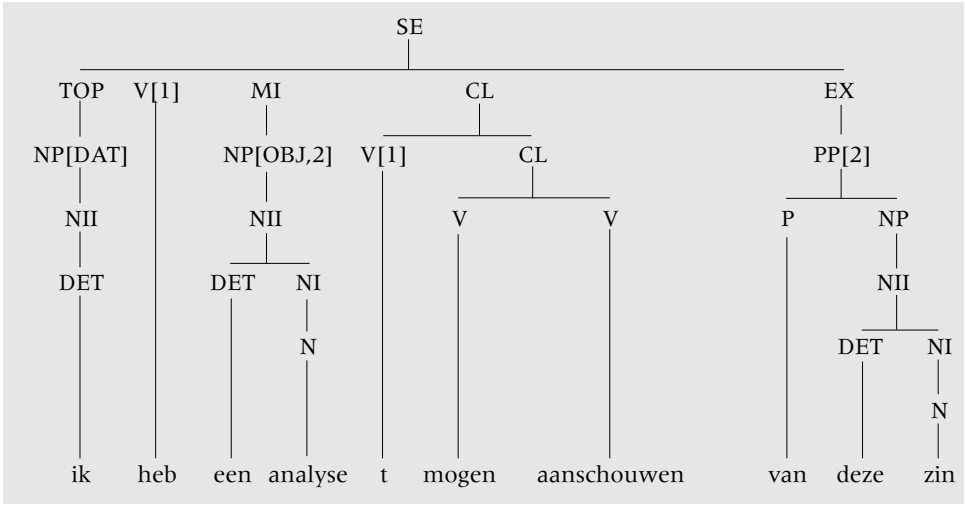
Tegelijk met de uitbouw van Amazon in de vorm van de Casusmodule vond er ook een vorm van afbraak plaats: om de complexiteit van het Amazonprogramma te reduceren werd de morfologische module uit het programma verwijderd en in een aparte module ervóór geschakeld (getiteld Amamorph). Tegenwoordig zouden we deze module als een losse *tagger* beschouwen: Amamorph analyseerde de elementen uit de zin tot een reeks van woordcategorieën, die de invoer vormde voor de nu puur syntactische Amazonparser.




*Figuur 3: Een dependentie-analyse van Casus uit 1980*

**13** De term *thematische rol* of *thetarol* was in 1980 nog niet *en vogue* in de generatieve grammatica.

**14** In het voorbeeld is DAT de datief of ondervindende persoon (experiencer) en OBJ is het object of de neutrale thematische rol. Een andere casusrol was AGE, de agens of handelende persoon.




*Figuur 4: Een moderne Casus-analyse*

## 4 Modularisering

Na 1980, toen het volledige Amazon-Casussysteem eenmaal operationeel was, vond een geleidelijk proces van modularisering plaats. De eerste betrof het Amazonprogramma. In 1983 schreef de studente Computerlinguïstiek Jenny Cals een doctoraalscriptie over de mogelijkheid om de Amazonparser te herformuleren in de vorm van een contextvrije herschrijfgrammatica die door een bestaande parser generator[15] automatisch kon worden omgezet in een parser.

De voordelen van deze operatie zijn evident: de linguïstische inhoud van de Amazonparser zit geheel in de contextvrije herschrijfgrammatica, en de parseeralgoritmiek wordt overgelaten aan de parser generator. Dat betekent dat taalkundigen en informatici min of meer onafhankelijk van elkaar kunnen werken aan de optimalisatie van hun eigen onderdelen.

De nieuwe Amazongrammatica beschreef overigens geen Nederlandse zinnen, maar structuren van woordcategorieën. Zoals gezegd was de morfologische analyse in handen van het Amamorph-programma, dat de tagging van de zin verzorgde. De lexicale ambiguïteit die Amamorph detecteerde, werd versleuteld in een zogeheten lexical lattice,[16] dat

**15** Een parser generator is een soort compiler die een formele grammatica omzet in een uitvoerbaar programma. De parser generator in kwestie was getiteld GRAMMA, ontwikkeld door ir. Hans Meijer van de afdeling Informatica van de KUN. In zijn dissertatie (Meijer 1986) geeft Meijer een uitvoerige beschrijving hiervan.

**16** Het Amamorph lexical lattice was als volgt gedefinieerd: de woorden uit de zin werden achter elkaar gerangschikt, elk woord geprefigeerd met al zijn mogelijke woordcategorieën (het woordje *dat* bijvoorbeeld werd geprefigeerd met de mogelijkheden *onderschikkend voegwoord* en *attributief* of *zelfstandig gebruikt demonstrativum*). Categorieën die meerdere woorden overspanden (*met behulp van*), werden genoteerd bij het eerste woord. Met een voorbeeld: in het geval van de voorzetseluitdrukking *met behulp van* werd bij het woord *met* de categorie *voorzetseluitdrukking* voorzien van een markering dat de volgende twee woorden daarbij inbegrepen waren.

de invoer vormde voor de Amazonparser. Deze construeerde op basis van de syntaxis het optimale pad door de lexicale mogelijkheden.

Het enige algoritme dat de transitie van computerprogramma naar formele grammatica niet overleefde, was het algoritme voor de werkwoordelijke eindgroep. Aangezien echter het aantal mogelijkheden in de praktijk eindig is (het Nederlands kent slechts een eindig aantal groepsvormende werkwoorden en ze mogen maar een eindig aantal keren in één werkwoordelijke groep voorkomen), werd voor dit onderdeel van de grammatica simpelweg een deelgrammatica voor een eindige taal ingelast. Pas in 1987 werd een manier gevonden om het oorspronkelijke algoritme in het toenmalige type formele grammatica terug te halen (cf. Coppen 1987 en Van Dreumel & Coppen (te versch.)).

Voordat in 1987 de deelgrammatica voor het werkwoordelijke cluster geheel werd herzien, was in 1985 de deelgrammatica voor de Noun Phrase al onder handen genomen. In Stoop (1985) wordt de NP-grammatica uit Coppen (1985) (later uitgebreider beschreven in Coppen 1991) geschikt gemaakt voor Amazon. Ook deze NP-grammatica is geconstrueerd aan de hand van een verwachtingsmodel: de gedachte is dat de NP-specifiers en premodifiers (zoals telwoorden, lidwoorden en adjectiva) de verdeling van naamval over de NP regelen. Een NP-initieel telwoord maakt een definiet lidwoord onmogelijk (**twee de aanwezigen*), tenzij dat exceptioneel gemarkeerd wordt met naamval (genitief *twee der aanwezigen*, of partitief *twee van de aanwezigen*). Een indefiniet lidwoord maakt een daaropvolgend telwoord onmogelijk (**'n ene aanwezige* of **zulke twee aanwezigen*), en verzwakt de verplichting om een verbogen adjectief te hebben (zie voor een uitgebreide uiteenzetting hiervan Coppen 1991). Deze NP-grammatica kon met succes in de bestaande grammatica worden ingebouwd.

De tweede belangrijke modularisering in het Amazon-Casussysteem vond plaats in 1989, toen ook het Casusprogramma herschreven werd tot een formele (transformationele) grammatica, die geïnterpreteerd werd door een separate algoritmiek. Op de taggingmodule na was nu het gehele systeem gesplitst in een zuiver taalkundig gedeelte en een informaticagedeelte.

Dat de taggingmodule in de loop der jaren enigszins onderbelicht is gebleven, werd nooit als een nadeel gezien. Immers, het ging hier niet zozeer om een toepassing van parsering, maar om prototypes. Daarnaast was de inbreng van de morfologische module zeer gering: de morfologische analyse bleef beperkt tot enkele productieve afleidingen (zoals werkwoordvervoeging en een aantal verbuigingen), die in feite eindig waren, en de tagger deed geen enkele poging om lexicale ambiguïteit op te lossen. Dat werd geheel aan de syntaxis overgelaten. Toen dan ook de parser generator in de jaren negentig de voorzieningen bood voor de opname van zeer grote lexica in de parser werd eenvoudigweg besloten om een groot lexicon van woordvormen op te nemen in plaats van een aparte morfologische module. Bij een efficiënte algoritmiek kost een lexicongrootte van 200000 in plaats van 100000 ingangen slechts één beslissingsstap extra.

## 5 Robuustheid

Na de omwerking van de deelgrammatica voor de werkwoordelijke groep in 1987 bleef het een zevental jaren stil op het Amazonfront. De parser fungeerde als een eerste stap in

de analyse van de Nederlandse zin, en het werk concentreerde zich op de Casusmodule. Zolang er sprake was van een onderzoeksinstrument was er ook weinig reden om aan Amazon te sleutelen. De parser produceerde wel eens een stuk of tien structurele analyses van een zin, maar de verwerking daarvan wierp geen technische problemen op en desnoods kon met de hand de gewenste analyse geselecteerd worden.

Intussen ontwikkelde ook de parser generator zich tot een volwassener instrument: het AGFL-systeem.[17] Er werden lexiconvoorzieningen toegevoegd en manieren om ambiguïteit te bestrijden. Zo konden regelalternatieven worden gemarkeerd als meer of minder waarschijnlijk, en sommige mogelijkheden konden als een soort "last resort" worden gemarkeerd: ze mochten alleen toegepast worden als andere mogelijkheden gefaald hadden.

In het doctoraalonderzoek van Erik Oltmans (Oltmans 1994) werden de mogelijkheden van deze nieuwe voorzieningen geëxploreerd. Oltmans onderscheidde allereerst de toevallige ambiguïteit van de structurele ambiguïteit. Toevallige ambiguïteit heeft een lexicale oorsprong. Bij substitutie van de woorden in de zin door niet-ambigue varianten verdwijnt ook de ambiguïteit. Met een voorbeeld: de zin *wij vieren feesten* is ambigu, maar bij vervanging van *vieren* door *tweeën* of vervanging van *feesten* door *feestjes* verdwijnt de ambiguïteit.

Van structurele ambiguïteit is sprake als een reeks van ondubbelzinnige lexicale categorieën meerdere syntactische analyses toelaat. Er zijn twee groepen van structurele ambiguïteiten:

- *Aanhechtingsconstructies*, waarbij het de vraag is op welk niveau van een constituent een daaropvolgende constituent moet worden aangehecht.
- *Transparante grensconstructies*, waarin onduidelijk is aan welke kant van de grens bepaalde constituenten moeten worden aangehecht.

Er zijn twee aanhechtingsconstructies: de aanhechting van mogelijke apposities (met name PP's), en de nevenschikkingconstructies. In een zin als *we hebben de verhalen van de buurman van je broer gehoord* is de veelvuldige ambiguïteit van de appositionele PP's manifest. Gaat het om de buurman van je broer, gaat het over de verhalen van de buurman, hebben we ze van de buurman gehoord of van je broer? Het is echter van belang om in te zien dat precies dezelfde syntactische ambiguïteit bestaat bij elke andere invulling van woorden met dezelfde categorie, ook als semantische of pragmatische factoren een van de mogelijkheden duidelijk bevoordelen. In een zin als *we hebben kinderen van de broer van je vrouw gezien* wordt de syntactische aanhechting van beide PP's sterk beïnvloed door de (semantische) neiging van de woorden *kinderen* en *broer* om een familierelatie te leggen met een *van*-PP, en de onwaarschijnlijkheid dat bij het werkwoord *zien* een bijwoordelijke *van*-PP is gerealiseerd. Aangezien Amazon geen toegang heeft tot informatie over subcategorisatie, laat staan tot diepere semantische of pragmatische informatie, is er binnen Amazon geen manier om uit alle mogelijkheden de juiste te kiezen.

Bij nevenschikkingsconstructies is de zaak nog problematischer, omdat er vaak ook

**17** AGFL staat voor Affix Grammars over Finite Lattices. Deze term slaat op het taalkundige formalisme, dat een vorm is van de tweeniveaugrammatica: een contextvrije herschrijfgrammatica, waarin de symbolen kunnen worden voorzien van kenmerken (affixen, attributen, features) die door middel van unificatie met de kenmerken van andere symbolen in verband kunnen worden gebracht.

samentrekkingskwesties meespelen, en omdat de aanhechting niet alleen rechtsrecursief maar ook linksrecursief kan zijn. Het standaardvoorbeeld is de nevenschikking *mannen en vrouwen en kinderen*, die op drie manieren kan worden gestructureerd. Amazon heeft geen middelen om hieruit de juiste te kiezen (als er al een juiste is).

De transparante grensconstructies zijn wat hun betekenis betreft wat minder manifest, maar ze vormen zeker geen kleiner probleem. Er zijn vier problematische gevallen. De eerste betreft de grens tussen het middenveld en het werkwoordelijk cluster, geïllustreerd in de volgende zin:

(3) Zouden [MI ze die film ook ingekleurd] [CL hebben]?
(4) Zouden [MI ze die film ook] [CL ingekleurd hebben]?

Het voltooid deelwoord *ingekleurd* kan een bepaling van gesteldheid zijn (voorbeeld (3)) of het hoofdwerkwoord van de zin (voorbeeld (4)). Die ambiguïteit is in dit voorbeeld reëel, in die zin dat het correspondeert met een duidelijk betekenisverschil. Het structurele karakter maakt echter dat in elke opeenvolging van dezelfde lexicale categorieën dezelfde ambiguïteit optreedt. Dus ook in:

(5) Zouden ze die film wel bekeken hebben?

Dat betekent dat een voltooid deelwoord aan het begin van de werkwoordelijke groep in principe altijd dubbelzinnig is: het kan altijd ook als bepaling van gesteldheid gezien worden. Dat deze analyse ook de historische oorsprong van de voltooid deelwoordconstructie is, is een schrale troost voor de parser: de structurele ambiguïteit dient in de contemporaine analyse geen enkel doel.

Een tweede transparante grensconstructie is zo mogelijk nog zinlozer. Vergelijk de volgende voorbeelden:

(6) Jan heeft [MI tijdens de pauze aan de vakantie] [CL gedacht]
(7) Jan heeft [MI tijdens de pauze] [CL gedacht] [EX aan de vakantie]
(8) Jan denkt [MI tijdens de pauze aan de vakantie] [CL]
(9) Jan denkt [MI tijdens de pauze] [CL] [EX aan de vakantie]

PP's kunnen zowel aan het einde van het middenveld optreden als aan het begin van het extrapositieveld. Dat is duidelijk te zien in de zinnen (6) en (7). Maar diezelfde twee mogelijkheden bestaan er ook in de zinnen (8) en (9), waarin het werkwoordelijke cluster oningevuld is. In dat geval is er voor Amazon geen enkele manier om uit te maken welk van de twee analyses de juiste is.

Een derde transparante grensconstructie betreft de grens tussen hoofdzin en beknopte bijzin:

(10) Jan beloofde maandag [SE om het gras te maaien]
(11) Jan beloofde [SE om maandag het gras te maaien]
(12) Jan beloofde maandag [SE het gras te maaien]
(13) Jan beloofde [SE maandag het gras te maaien]

Ook hier kan een zinsdeel (*maandag*) links of rechts van een grens staan. Deze grens kan gemarkeerd zijn door het voegwoord (*om*), maar als het voegwoord achterwege blijft en het werkwoordelijke cluster in de hoofdzin is leeg, dan is er voor Amazon geen manier om de constituenten op deze grens in de goede zin te plaatsen, zelfs niet als de subcategorisatieframes van de werkwoorden één van de mogelijkheden zouden uitsluiten. De zinnen (12) en (13) vormen dan ook een structurele ambiguïteit.

Een laatste transparante grensconstructie betreft de volgende:

(14) Ze hebben [NP die twee oude sigaren] gegeven
(15) Ze hebben [NP die twee] [NP oude sigaren] gegeven
(16) Ze hebben [NP die] [NP twee oude sigaren] gegeven
(17) Ze hebben [NP die twee oude] [NP sigaren] gegeven

Hier hebben we niet zozeer te maken met constituenten die aan weerszijden van een grens kunnen staan, maar eerder met de onduidelijkheid van de grens zélf. Het Nederlands kent het kale meervoud: meervoudige substantieven kunnen zonder lidwoord of telwoord een NP vormen. Daarnaast kan bij specificatie of modificatie met demonstrativa en adjectiva de kern van de NP worden weggelaten. De combinatie van die twee eigenschappen levert een uiterst productieve structurele ambiguïteit op. De vier analyses voor dezelfde woordreeks in (14) tot en met (17) zijn technisch gezien nog niet eens de enige mogelijkheden, omdat de reeks *die twee oude sigaren* ook zou kunnen worden geanalyseerd als een opeenvolging van drie NP's:[18]

(18) [NP die] [NP twee oude] [NP sigaren]

Dit soort analyses mogen onwaarschijnlijk zijn, het is onduidelijk op welke syntactische gronden ze kunnen worden uitgesloten. Uiteraard zijn er een aantal factoren die deze ambiguïteiten in het dagelijkse taalgebruik onderdrukken (zoals de subcategorisatie van de omringende werkwoorden, de collocaties, of de menselijke parseerstrategie om zo groot mogelijke constituenten te maken: de *late closure* strategie), maar geen van deze factoren behoort tot het domein van de structurele syntaxis.

Ook al bieden robuustheidsvoorzieningen in het AGFL-systeem in principe de mogelijkheden om in geval van structurele ambiguïteit een keuze af te dwingen, aangezien de benodigde informatie voor het maken van zo'n keuze niet beschikbaar is, kan de correctheid van de keuze nooit gegarandeerd worden. Er zijn vier manieren om aan deze patstelling te ontsnappen:

- Geen keuze maken: alle mogelijkheden als analyse opleveren;
- De benodigde informatie voor het maken van de correcte keuze aan de grammatica toevoegen;
- Een ondergespecificeerde analyse afleveren, waarin meerdere keuzes in één structuur vervat zijn;
- Een willekeurige, dus soms verkeerde keuze maken.

**18** Ook andere combinaties zijn mogelijk. Bovendien is onduidelijk op welke syntactische gronden een analyse in *vier* NP's zou moeten worden uitgesloten.

De eerste strategie ligt het meest voor de hand. In feite is dat voor een groot deel de manier waarop de Amazonparser tot 1994 te werk ging. Helaas geeft deze strategie in theorie aanleiding tot de zogeheten *combinatorische explosie*: het aantal analyses neemt exponentieel toe met de lengte van de zin. Snellere hardware of meer geheugen kunnen de drempel voor een werkbaar aantal analyses marginaal verleggen, principieel is deze oplossing ondeugdelijk.

De tweede strategie is ook al ondoenlijk, aangezien op den duur zeer verfijnde informatie nodig is voor desambiguering. Allerlei kennis van de wereld bepaalt in een concrete taalgebruikssituatie de voorkeursanalyse. Ook al zou het inbouwen van deze informatie in de grammatica in principe mogelijk zijn, dan nog is het onduidelijk waar we deze kennis vandaan moeten halen.

De derde strategie lijkt heel doenlijk: door het geven van een ondergespecificeerde analyse blijven alle mogelijkheden intact, en wordt de keuze uitgesteld tot latere modules die meer, of andere, informatie beschikbaar hebben.

De laatste mogelijkheid, een verkeerde keuze maken, lijkt absurd: ook al zou het mogelijk zijn om op basis van waarschijnlijkheid in veel gevallen de juiste analyse op te leveren, wat voor nut kan het hebben om in een substantieel aantal gevallen de verkeerde keuze te maken en zo verdere analyse onmogelijk te maken?

Toch is het maken van een willekeurige keuze niet zo absurd als het lijkt. Waar het om gaat is dat een mogelijk verkeerde keuze gemaakt is in een structurele context die door latere modules herkenbaar is. Met een voorbeeld: bij een opeenvolging van NP en PP kan de parser de PP aanhechten als rechterzuster dan wel meest rechtse dochter van de NP. Als een latere module, die zich bijvoorbeeld met subcategorisatie bezighoudt, beide constructies aanmerkt als mogelijk verkeerd, en een reparatiecomponent bevat die de andere mogelijkheid indien gewenst kan exploreren, is geen enkele mogelijkheid afgesneden. Welk van de mogelijkheden aanvankelijk wordt gekozen, is nu van secundair belang.

Waarom zou deze strategie beter zijn dan het ontwikkelen van een ondergespecificeerde analyse? Het antwoord is dat het eenvoudiger is. De ondergespecificeerde analyse is een extra voorziening die als enige doel heeft het markeren van de plaatsen waar later nadere specificatie moet plaatsvinden. Maar als we die plaatsen toch al kunnen aanwijzen (bijvoorbeeld als elke opeenvolging van NP en PP) is zo'n speciale voorziening dus onnodig. Waarom zou een "verkeerde" structuur dan slechter zijn dan geen structuur?

Dit is de gedachte die in het doctoraalonderzoek van Oltmans gevolgd is: het is bij structurele ambiguïteit niet nodig dat Amazon de juiste keuze maakt of een aparte representatie verzint, als de parser maar geen twee in principe gelijkwaardige analyses oplevert.

Deze robuustheidsstrategie is in latere versies van Amazon verder uitgewerkt. Voor alle structurele ambiguïteiten zijn praktische keuzes gemaakt, die wel ingegeven zijn door een logische gedachte, maar die net zo goed anders hadden kunnen uitvallen:[19]

- Bij aanhechting van PP's is gekozen voor een zo hoog mogelijke aanhechting. Alle PP's worden zo mogelijk los gegenereerd. Als de PP's appositioneel zijn, dan moet dat in latere modules maar blijken.

**19** Die keuzes zijn dus wel gerelateerd aan een rudimentair begrip van waarschijnlijkheid, maar in wezen zijn ze willekeurig.

- Nevenschikkingen worden beschreven net als PP's: aparte constituenten met een voegwoord (of komma) als kern worden hoog aangehecht.
- Voltooide deelwoorden worden indien mogelijk onder de werkwoordgroep gerekend.
- Bij een lege werkwoordgroep worden zoveel mogelijk constituenten onder het middenveld gegenereerd.
- Bij beknopte bijzinnen worden lege middenvelden ontmoedigd.
- Bij opeenvolging van hoofdloze en kale meervoud-NP's wordt ernaar gestreefd zo groot mogelijke NP's te maken.

Geen van deze keuzes is principieel: voor PP's en nevenschikking geldt een *early closure*, *maximal attachment* strategie, waarbij maximale projecties zo snel mogelijk worden gesloten en zo hoog mogelijk worden aangehecht. Voor het middenveld en reeksen van NP's is gekozen voor een *late closure* strategie, waarbij de constituent of het veld zo lang mogelijk open blijft.

Met de verwijdering van structurele ambiguïteit daalde het aantal analyses dat de Amazonparser gemiddeld per zin opleverde natuurlijk dramatisch. Dat maakte het interessant om te bezien of Amazon ook inzetbaar zou zijn als technologisch instrument, voor de analyse van concreet taalgebruik in plaats van zorgvuldig geselecteerde modelzinnen.

Hiertoe werd de grammatica voorzien van een groot lexicon (ongeveer 325000 woordvormen) en een aparte *last resort* deelgrammatica, die bij ongrammaticale zinnen zo goed mogelijke deelanalyses moest opleveren. Het is deze organisatie die in de huidige versie van Amazon nog steeds in gebruik is.

## 6. De stand van zaken

In de 27 jaar ontwikkeling van Amazon zijn een drietal trends aan te wijzen:

- **Modularisering**: in de loop der jaren is Amazon steeds verder gemodulariseerd. Sommige onderdelen (zoals tagging en morfologische analyse) zijn geheel verwijderd uit de grammatica, andere (werkwoordgroep, NP) zijn alleen afgezonderd van de andere grammatica's.
- **Scheiding van Algoritme en Taalkunde**: de taalkundige beschrijving is neergelegd in een apart formalisme, dat niets met de algoritmiek van de parser te maken heeft.
- **Robuustheid**: door de jaren heen heeft Amazon zich steeds meer ontwikkeld van een puur taalkundig "proof of principle" prototype tot een meer technologisch instrument voor de analyse van concreet taalgebruik. Dat blijft voorlopig nog beperkt tot schriftelijk taalgebruik, maar er worden al enkele experimenten gedaan met de analyse van getranscribeerde spraak.

In de loop van haar geschiedenis is Amazon verschillende malen *from scratch* herschreven; daarbij is de terminologie enkele malen gemoderniseerd, en zijn "dichtgeslibde" deelgrammatica's opnieuw opgesteld. Belangrijke herschrijvingen hebben plaatsgevonden in 1983 (Jenny Cals), 1985 (Albert Stoop), 1987 (Peter-Arno Coppen), 1994 (Erik Oltmans) en 1997 (Simon van Dreumel). Voor 2002 staat een nieuwe versie op het programma.

1. Cathy zag hen wild zwaaien.
2. haar vader stak zijn duim omhoog alsof hij wilde zeggen: het komt wel goed, joch.
3. haar moeder kleefde bijna tegen het autoraampje aan.
4. haar neus werd platgedrukt en leek op een jonge champignon.
5. Cathy zag de BMW langzaam verdwijnen tot hij niet meer was dan een zilveren schijnsel tussen de bomen en struiken.
6. ze veegde de tranen uit haar ooghoeken, tilde haar twee koffers op en begaf zich in de richting van het landhuis.
7. de oprijlaan was niet meer dan een hobbelige zandstrook die zich voortslingerde tussen de hoge grijze boomstammen.
8. de middagzon hing klein tussen de takken en de schaduwen van de wolken drentelden over het gras.
9. het had een prachtige dag kunnen zijn in Londen.
10. ze had met haar moeder kunnen gaan winkelen, zwemmen of terrassen.
11. dat werkwoord had ze zelf uitgevonden.
12. het hoorde bij de warme zomerdag die ze ginds achter had gelaten.
13. ze hadden languit naast elkaar op de strandstoelen kunnen gaan liggen.
14 zij zou mams rug ingewreven hebben en mam de hare.
15. of ze had gewoon met haar vriendinnen rond kunnen slenteren in de buurt van Trafalgar-Square.
16. elk jaar in het hoogseizoen trokken daar massa's toeristen voorbij, hun fototoestel in de aanslag, pratend, gillend en lachend in de vreemdste talen.
17. het was een spel geworden: zij en haar vriendinnen kozen iemand uit en probeerden zijn of haar nationaliteit te raden.
18. het meisje dat vijf keer juist raadde werd getrakteerd op ijs.

*Figuur 5: Het minicorpus van testzinnen*

Op dit moment lopen er drie Amazonprojecten:

- Het AIO-project van Simon van Dreumel, waarin geprobeerd wordt om het einde van het middenveld meer structuur te geven. Amazon beschouwt het middenveld als een willekeurige reeks van maximale projecties en partikels, maar het is duidelijk dat de mogelijkheden beperkter zijn. Zo zal een werkwoordpartikel nooit vóór een maximale projectie staan (Van Dreumel 1997).
- Het AIO-project van Carla Schelfhout, dat een poging doet om intercalaties[20] in de Amazongrammatica op te nemen (Schelfhout 1999).
- De doctoraalprojecten van Jan Smeets en Bram Elffers, waarin een alternatieve versie van Amazon wordt geschreven die een beperkte vorm van subcategorisatie omvat (Smeets 2002, Elffers 2002).

Net afgerond is het doctoraalproject van Anouk Gerrits, waarin de clitische groep en de partikelreeksen aan het begin van het middenveld worden beschreven (Gerrits 2001).

**20** Intercalaties zijn discontinuïteiten in de vorm van ingelaste tussenzinnen of andere constituenten, gekenmerkt door een breuk in de intonatie.

Het Amazonproject kent ook een website waar de laatste ontwikkelingen worden aangekondigd.[21]

## 7 De prestaties

Voor de LOT winterschool 2001 hebben de betrokken parseronderzoekers een minicorpus van taaluitingen gekozen uit het voorgelezen materiaal van het CGN-corpus. Het betreft dus in feite schriftelijk materiaal, waarin de interpunctie naar eigen inzicht is aangebracht. Het corpus omvat 18 zinnen met een zinslengte van 5 tot 23 woorden (zie figuur 5).

De onderzoekers mochten hun parsers op het corpus "tunen", hetgeen in elk geval zou inhouden dat het lexicon kon worden aangepast (geen van de lexica zou het werkwoord *terrassen* bevatten, dat door de hoofdpersoon van het verhaal was "uitgevonden").

Hoe moet je een parser evalueren, of meerdere parsers vergelijken? Voor de hand ligt om de uitvoer van de parser te vergelijken met een "gouden standaard": een handmatig uitgevoerde of gecontroleerde "correcte" analyse. Dat brengt echter, zo constateerden we tijdens de winterschool, een aantal problemen met zich mee. Vooreerst: wat is eigenlijk de "correcte" analyse? Een generatieve analyse in de Chomskyaanse minimalistische theorie, compleet met sporen en lege categorieën? Een klassieke zinsontleding, een predikaatlogische formule? Maar zelfs als we daaruit een keuze maken, bijvoorbeeld als minst controversiële de klassieke zinsontleding, dan levert de eerste de beste zin al problemen op: moeten we *hen* en *zwaaien* als twee lijdende voorwerpen analyseren (de gangbare opvatting), of toch maar *hen zwaaien* als beknopte lijdend voorwerpszin met *hen* als onderwerp? Of maken we van *zien* een nieuw soort hulpwerkwoord, waardoor we van *hen* het enige lijdend voorwerp kunnen maken?

Het is geen toeval dat de eerste zin al problemen oplevert. Ook over de tweede zin kan gediscussieerd worden: is *omhoog* een werkwoordpartikel (ik zou argumenteren van niet) of een bijwoordelijke bepaling van richting? Is *het* een gewoon onderwerp of loos (daar is discussie over mogelijk, maar ik zou denken het laatste)? Het zal duidelijk zijn: de klassieke zinsontleding mag dan de indruk wekken een relatief onproblematische analysemethode te zijn, in de praktijk levert bijna elke zin discussie op.

Daarbij komt dat vergelijking van de uitvoer van een parser met een klassieke zinsontleding in sommige gevallen bepaald oneerlijk zou zijn: de parser Delilah bijvoorbeeld levert een predikaatlogische formule die aanmerkelijk dieper is dan een gewone zinsontleding. Voor deze diepte wordt Delilah in een vergelijking gestraft. Er zou een extra inspanning nodig zijn om de resultaten van Delilah terug te rekenen naar een klassieke ontleding. Aan de andere kant van het spectrum geldt voor Amazon dat bepaalde onderscheidingen bewust achterwege zijn gelaten. Vergelijking met een analyse waarin die onderscheidingen wél zijn gemaakt, zou suggereren dat Amazon ze eigenlijk wel had moeten maken. Dat geeft dan wel aan hoever de Amazonanalyse verwijderd is van de gekozen standaard, maar niet van de *beoogde* analyse.

Tijdens de winterschool kwamen we er niet helemaal uit, en bleef de evaluatie van de

**21** Om voor de hand liggende redenen kan het adres natuurlijk niet amazon.nl of amazon.com zijn. Daarom is het <http://lands.let.kun.nl/amazon>.

parsers beperkt tot het beoordelen van incidentele parseerresultaten. Toch is in het geval van Amazon een gerichtere evaluatie wel mogelijk, al dienen bij iedere evaluatie kanttekeningen gemaakt te worden.

Allereerst hebben we de zinnen van het testcorpus handmatig geanalyseerd volgens de Amazondescriptie. Dat wil zeggen: constituenten worden benoemd met Amazonlabels, alleen structurele velden worden onderscheiden, en de aanhechtingsproblematiek wordt handmatig opgelost door hoge aanhechting. Deze analyse wordt vergeleken met de Amazonresultaten. Wat we daarmee testen is hoe goed de parser presteert *volgens de bedoeling*. Zoals gebruikelijk meten we twee grootheden: de *precision* (het aantal correct benoemde constituenten gedeeld door het totale aantal benoemde constituenten) en de *recall* (het aantal correct benoemde constituenten gedeeld door het totale aantal constituenten dat benoemd had moeten worden). Beide grootheden worden op twee manieren gemeten: alleen op woordgroepsniveau (waarbij de benoemingslabels en losse woorden niet meetellen), en volledig (woord of woordgroep inclusief benoemingslabel). De gedachte hierachter is dat de eerste maat aangeeft hoe goed de woordgroepverdeling is, en de tweede hoe dicht de Amazonparser het doel nadert.

Soms geeft de Amazonparser meer dan één analyse (in één geval 6, in 3 gevallen 2). In die gevallen worden alle analyses meegerekend. Dit zou de recall gunstig kunnen beïnvloeden (immers de kans dat de goede benoemingen erbij zitten wordt groter), maar de precision moet kleiner worden (er zullen zeker verkeerde benoemingen bij zitten). Dubbele benoemingen tellen uiteraard maar één keer. In figuur 6 staan de resultaten van de Amazonparsering.

| Zin | Precision (constituent) | Recall (constituent) | Precision (totaal) | Recall (totaal) |
|---|---|---|---|---|
| 1 | 1. | 1. | 1. | 1. |
| 2 | 1. | 1. | 1. | 1. |
| 3 | 0.67 | 0.67 | 0.83 | 0.83 |
| 4 | 1. | 1. | 1. | 1. |
| 5 | 1. | 1. | 0.94 | 0.95 |
| 6 | 1. | 1. | 1. | 1. |
| 7 | 0.63 | 0.91 | 0.78 | 0.93 |
| 8 | 1. | 1. | 0.95 | 0.91 |
| 9 | 1. | 1. | 1. | 1. |
| 10 | 1. | 1. | 1. | 1. |
| 11 | 1. | 1. | 1. | 1. |
| 12 | 1. | 1. | 0.90 | 0.92 |
| 13 | 1. | 1. | 1. | 1. |
| 14 | 1. | 1. | 1. | 1. |
| 15 | 1. | 1. | 0.98 | 1. |
| 16 | 0.56 | 0.82 | 0.69 | 0.88 |
| 17 | 1. | 1. | 1. | 1. |
| 18 | 1. | 1. | 1. | 1. |
| **gemiddeld** | **0.87** | **0.91** | **0.89** | **0.91** |

*Figuur 6: Evaluatie van Amazon op minicorpus*

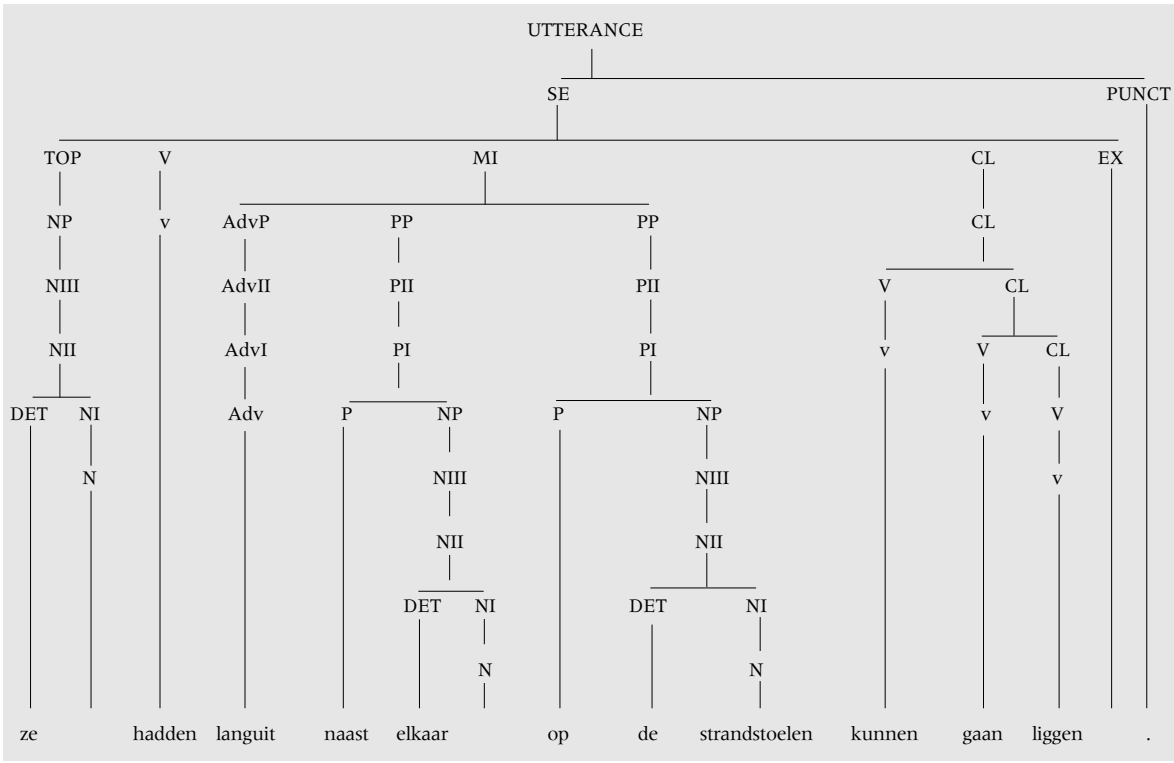
Duidelijk is dat Amazon moeite heeft met zin 16, die eindigt in een nevenschikking van bepalingen van gesteldheid (*hun fototoestel in de aanslag, pratend, gillend en lachend in de vreemdste talen*) waarvan het nog maar de vraag is hoe ze ingebed zitten. In de handmatige, "correcte" analyse staan ze als vier zinsdelen nevengeschikt in de uitloop van de zin, hoewel ook een nevenschikking van twee bepalingen van gesteldheid verdedigbaar zou zijn met een nevenschikking van drie tegenwoordig deelwoorden in het tweede conjunct. Amazon geeft hier twee mogelijkheden, waardoor met name de precision daalt. Bovendien heeft de parser moeite met het kiezen van de juiste aanhechting voor de nevenschikking. In feite wordt de totale juiste analyse niet gevonden. Dat de recall nog tamelijk hoog is ligt aan het feit dat de relatief onproblematische kleinere eenheden (losse NP's en minor constituents) allemaal goed gaan.

Men kan opmerken dat een dergelijke evaluatie geen goede indruk geeft van de prestatie van de Amazonparser in vergelijking tot een uiteindelijke gewenste parsering, waarbij alle constituenten op het juiste niveau aangehecht zijn en correct benoemd naar hun zinsdeelfunctie. Ook dat is eenvoudig handmatig te testen. Met name de handmatige correctie van de analyse naar aanhechting van nevenschikking en nabepalingen is eenvoudig. In figuur 7 staan de resultaten van deze berekening.

| Zin | Precision (constituent) | Recall (constituent) | Precision (totaal) | Recall (totaal) |
|---|---|---|---|---|
| 1 | 1. | 1. | 1. | 1. |
| 2 | 1. | 0.9 | 0.95 | 0.87 |
| 3 | 0.67 | 0.67 | 0.83 | 0.83 |
| 4 | 1. | 1. | 1. | 1. |
| 5 | 0.86 | 1. | 0.94 | 0.83 |
| 6 | 0.94 | 0.88 | 0.95 | 0.94 |
| 7 | 0.63 | 0.91 | 0.78 | 0.93 |
| 8 | 1. | 1. | 0.95 | 0.91 |
| 9 | 1. | 1. | 1. | 1. |
| 10 | 1. | 1. | 1. | 1. |
| 11 | 1. | 1. | 1. | 1. |
| 12 | 1. | 1. | 0.90 | 0.92 |
| 13 | 1. | 1. | 1. | 1. |
| 14 | 1. | 1. | 1. | 1. |
| 15 | 0.90 | 0.90 | 0.90 | 0.90 |
| 16 | 0.56 | 0.82 | 0.79 | 0.88 |
| 17 | 0.61 | 0.85 | 0.69 | 0.88 |
| 18 | 1. | 1. | 1. | 1. |
| **gemiddeld** | **0.85** | **0.88** | **0.87** | **0.89** |

*Figuur 7: Evaluatie gecorrigeerd naar aanhechting*

De daling van precision en recall blijkt niet zo dramatisch, maar dat lijkt een te rooskleurige voorstelling van zaken. In het minicorpus komen relatief weinig aanhechtingsproblemen voor.

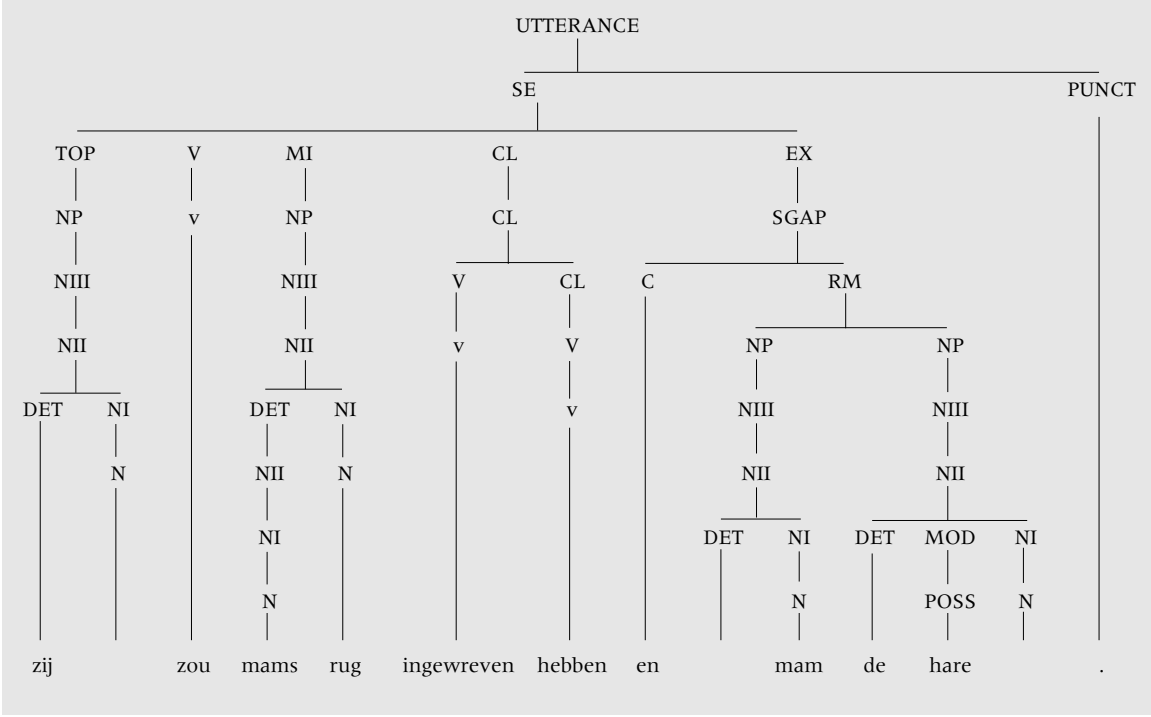



*Figuur 8: Amazonanalyse van zin 13*

Wanneer we ook nog de benoeming volgens de klassieke ontleding in de codering zouden opnemen, blijven de getallen voor precision en recall op constituentniveau uiteraard gelijk (ervan uitgaande dat de structuralistische Amazonanalyse alleen onderspecificeert voor functionele benoeming). Zowel totale precisie als totale recall zullen echter dalen, afhankelijk van wat we zouden toevoegen. Moeten we *haar* in zin (3) als bijvoeglijke bepaling bij *moeder* benoemen, of is dat in de structuur al uitgedrukt?

In plaats van deze verfijning toe te passen, met zijn vele onzekere parameters, geef ik in een tweetal voorbeelden een indruk van de Amazonparsering. In figuur 8 staat de analyse van zin 13, met een mooie werkwoordelijke eindgroep. De werkwoordgroep *hadden kunnen gaan liggen,* inclusief IPP-constructie, wordt door Amazon correct herkend. De PP's *naast elkaar* en *op de strandstoelen* worden naast elkaar onder het middenveld gegenereerd. Dat is niet gek, al is het nog maar de vraag wat in deze zin de beste analyse is. Moeten *languit* en *naast elkaar* niet als één bepaling van gesteldheid gezien worden?

In figuur 9 (zin 14) komt nevenschikking met samentrekking ("gapping") voor. Het tweede conjunct bestaat uitsluitend uit overblijfselen (remnants) van de samentrekking. Amazon geeft ook hier de correcte analyse (en overigens wordt het voltooid deelwoord *ingewreven* onder CL geplaatst en niet onder MI, zoals boven uitgelegd). De volledige analyses van het minicorpus zijn na te zien op de Amazon-website.




*Figuur 9: Amazonanalyse van zin 14*

Dit resultaat moet natuurlijk met enige scepsis bezien worden. Zoals gezegd is Amazon een parser met een bescheiden doelstelling. Door met opzet problematische keuzes buiten de analyse te houden, moet het resultaat uiteraard gunstig beïnvloed worden. En per slot van rekening is de parser "getuned" op de invoer. Niettemin werd tijdens de gezamenlijke sessie duidelijk dat Amazon wel degelijk een robuuste parser genoemd kan worden: Amazon was de enige van de deelnemende parsers die in staat was om ter plekke een onvoorbereid stuk tekst te analyseren met een acceptabel resultaat.

## 8 De toekomst van de Oude Dame

Wat is de toekomst van de Amazonparser? Wat heeft oppervlakteparsering voor nut in een wereld waar de computers steeds krachtiger worden? Heeft het niet meer zin om in te zetten op ambitieuzere parsers? Wat is de theoretische status van een oppervlakteanalyse?

In de loop der jaren is er binnen de Amazon-onderzoeksgroep herhaaldelijk discussie geweest over de reikwijdte van de Amazonanalyse. De verleiding om Amazon uit te breiden met allerlei diepere vormen van analyse is met de modernisering van hardware en software altijd groot geweest. Toch is er wel degelijk een theoretische basis voor de opper-

vlakteparsering zoals Amazon die levert. Een goed voorbeeld is het verschil tussen de volgende zinnen:

(19) Ik dacht dat Jan Karel Marie de hond de krant uit de bus heeft helpen leren laten halen.
(20) Ik dacht dat Jan de hond de krant uit de bus heeft helpen leren laten halen
(21) Ik dacht dat Jan Karel Marie de hond de krant uit de bus heeft helpen geleerd laten halen

Het is vrijwel onmogelijk om zonder pen en papier de grammaticaliteit van zin (19) te bevestigen. Hoewel grammaticaal, doet de constructie blijkbaar een te groot beroep op het werkgeheugen van de menselijke parser. Ook de ongrammaticaliteit in (20) is vrijwel ondetecteerbaar. In scherp contrast daarmee is zin (21) onmiddellijk herkenbaar als een foute zin. Blijkbaar is dit een ongrammaticaliteit die dichter bij de oppervlakte van de zin blijft. Een grammaticale theorie zou dit moeten verantwoorden.

De Amazonparsering detecteert zin (21) als een foute zin (dat wil zeggen: Amazon parseert deze zin met de robuustheidsgrammatica en beschouwt hem als een ellips), en produceert normale analyses van de andere twee zinnen. Daarmee is een reëel onderscheid in de parsering tot uitdrukking gebracht. De ongrammaticaliteit van zin (20) moet in de Amazonvisie door andere modules dan syntactische parsering gedetecteerd worden.

Deze bescheiden doelstelling van de Amazonparser is geen ontkenning van de noodzaak tot ambitieuzere parseerinstrumenten. Uiteraard is voor de meeste toepassingen een dieper inzicht noodzakelijk in de structuur en de betekenisverhoudingen van de taaluiting. Zeer waarschijnlijk moeten daar ook kwesties met betrekking tot de tekstuele en buitentekstuele context in meegenomen worden. Waar Amazon echter voor pleit is om oppervlakteparsering als een aparte module aan het begin van een langer analysetraject te handhaven, eventueel voorafgegaan door een module die de toekenning van woordcategorieën verzorgt. Oppervlakteparsering is een theoretisch heel goed verdedigbaar analyseniveau, en het is heel goed te doen met acceptabele resultaten. De beperkingen van oppervlakteparsering dienen niet te worden opgelost door de parser te compliceren, maar door separate verrijkings- of herstelmodules. Alleen op die manier blijft de organisatie van de analyse helder, en wordt het gevaar van de combinatorische explosie, met alle gevolgen voor de analysetijden van dien, efficiënt bestreden.

## Bibliografie

**Bakel, Jan van (1975).** *Automatische zinsontleding met de computer*. Interne publicatie KU Nijmegen.

**Cals, Jenny (1983).** *Een contextvrije niet-linksrecursieve grammatica die dezelfde haakjesstructuur produceert als 'AMAZON'*. Doctoraalscriptie KU Nijmegen.

**Coppen, P.A. (1985).** De aard van het quantitatieve *er*. *De Nieuwe Taalgids* 78, 149-163.

**Coppen, P.A. (1987).** Het AMAZON-algoritme voor werkwoordelijke eindclusters. *Gramma* 11, 1-17.

**Coppen, P.A. (1991).** *Specifying the Noun Phrase*. Dissertatie KU Nijmegen.

**Dreumel, S. van & P.A. Coppen (te versch.).** Surface analysis of the Verbal Cluster in Dutch. *Linguistics*.

**Dreumel, S. van (1997).** A Robust parser for Dutch Sentences, abstract PhD project, <http://lands.let.kun.nl/~dreumel/PhD_project.nl.html>.

**Elffers, A. (2001).** Transducing Dutch utterances into Head/Modifier pairs. *Proceedings of the 2nd AGFL Conference*, <http://www.cs.kun.nl/agfl/workshop2/bramelf.pdf>.

**Fillmore, C (1968).** The Case for Case. In: E. Bach & R.J. Harms (eds) *Universals in Linguistic Theory*, New York: Holt, Rinehart & Winston.

**Gerrits, A. (2001).** *Het begin van het middenveld*. Doctoraalscriptie KU Nijmegen.

**Meijer, H. (1986).** *Pro Grammar: A Translator Generator*. Dissertatie KU Nijmegen.

**Oltmans, J.A. (1994).** *Amazon in AGFL: een contextvrije herschrijfgrammatica voor de structurele module van het AMAZON/CASUS-systeem, beschreven in het AGFL-formalisme*. Doctoraalscriptie KU Nijmegen.

**Oostdijk, N. & H. van Halteren (2002).** De grammaticale annotatie van tekstcorpora, *Nederlandse Taalkunde* 7, 175-181.

**Rijpma, E. & F.G. Schuringa (1968).** *Nederlandse spraakkunst*, bewerkt door Jan van Bakel. Groningen: Wolters-Noordhoff.

**Schelfhout, C. (1999).** *Intercalaties*, <http://lands.let.kun.nl/amazon/Algemeen/carla.htm>.

**Smeets, J. (2002).** A subcategorisation model for Dutch and its implementation in AGFL, *Proceedings of the 2nd AGFL Conference*, <http://www.cs.kun.nl/agfl/workshop2/smeets.pdf>.

**Stoop, A. (1985).** *De implementatie van de NP-Coppen in Amazon en Casus*. Doctoraalscriptie KU Nijmegen.

# Syntactische annotatie voor het Corpus Gesproken Nederlands (CGN)

TON VAN DER WOUDEN, HELEEN HOEKSTRA, MICHAEL MOORTGAT,
BRAM RENMANS EN INEKE SCHUURMAN*

**Abstract**

**The paper discusses the syntactic annotation for the Spoken Dutch Corpus, a Dutch/Flemish cooperation project to build an annotated corpus of about one thousand hours of continuous speech, which amounts to 10 million words. After a brief introduction to the project, we discuss the kind of syntactic annotations we envisage (dependency structures) and the way they are created (semi-automatically). We mention some peculiarities of spoken language, and we finish with a discussion of some of the kinds of questions the corpus may help answering.**

## 1 Inleiding

Dit artikel besteedt aandacht aan de syntactische annotatie ten behoeve van het Corpus Gesproken Nederlands (in het vervolg meestal CGN). In de tweede paragraaf worden doel en opzet van het CGN besproken, alsmede de plaats van de syntactische annotatie daarin. In de derde paragraaf bespreken we het soort syntactische structuren dat het CGN oplevert. In de tamelijk technische vierde paragraaf gaan we in op de uitgangspunten van de syntactische analyse, en in de vijfde op de praktische implementatie van het proces. In de zesde paragraaf bespreken we een aantal specifieke problemen verbonden aan het ontleden van gesproken taal. In de zevende en laatste paragraaf tenslotte worden enkele voorbeelden behandeld van typen vragen die taalkundigen altijd over het Nederlands hadden willen stellen en die nu met behulp van een syntactisch geannoteerd corpus van het gesproken Nederlands niet alleen gesteld, maar misschien ook daadwerkelijk beantwoord kunnen worden.[1]

* Adres: Van der Wouden, Hoekstra en Moortgat: UiL-OTS, Trans 10, 3512 JK Utrecht, Renmans en Schuurman: K.U.Leuven, Centrum voor Computerlinguïstiek, Maria-Theresiastraat 21, 3000 Leuven.

1 Deze publicatie is tot stand gekomen in het kader van het project "Corpus Gesproken Nederlands" met financiële steun van de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) en de Vlaamse Overheid. Dit artikel is ten dele een vertaling annex samenvatting van gedeelten van Hoekstra et al. (2001a, 2001b).

## 2 Het Corpus Gesproken Nederlands

Het Corpus Gesproken Nederlands is een samenwerkingsproject van een aantal Nederlandse en Vlaamse universiteiten (Goedertier et al. 2000, Oostdijk 2000a, 2000b). Het project, dat wordt gefinancierd door NWO en FWO en beheerd door de Taalunie, is begonnen in juni 1998 en heeft een looptijd van vijf jaar. Het einddoel is een geannoteerd corpus van ongeveer duizend uur lopende spraak, wat neerkomt op zo'n tien miljoen woorden.[2]

Het CGN is bedoeld als een bron, een nieuw soort bron, van informatie voor taalkundig onderzoek en voor taal- en spraaktechnologie. Om deze verschillende doelgroepen optimaal te kunnen bedienen, wordt het corpusmateriaal verzameld in uiteenlopende communicatieve situaties, waaronder spontane dialogen, telefoongesprekken, vraaggesprekken, discussies, debatten, lezingen, nieuwsuitzendingen en voorgelezen literatuur. Tweederde van het materiaal is afkomstig uit Nederland, eenderde uit het Nederlands sprekende gedeelte van België.[3] Als het corpus voltooid is, zal het de grootste en meest gevarieerde verzameling gesproken Nederlands zijn die tot dusver bijeengebracht is. Tussentijdse versies (die inmiddels al meer dan 100 CD-ROMs vullen) worden gedistribueerd via ELRA.[4]

Het project beoogt verschillende niveaus en typen van annotatie aan te bieden. Het gehele corpus wordt orthografisch getranscribeerd en taalkundig ontleed: ieder woord krijgt een woordsoort toegekend.[5] Een representatieve selectie van zo'n tien procent van de spraakdata – het zogenoemde 'kerncorpus' – wordt voorzien van een brede fonetische transcriptie en van een syntactische annotatie. Bovendien ontvangt een kwart van het kerncorpus, in totaal dus zo'n 250.000 woorden, een prosodische annotatie.[6] In deze bijdrage gaan we vooral in op de syntactische annotatie, dus op de redekundige ontleding.

## 3 De syntactische analyses van het CGN

Uitgangspunt voor de ontleding is de taalkundig ontlede orthografische transcriptie (dus niet het ruwe spraaksignaal – zo ver is de computationele taalkunde nog niet).[7] Het materiaal is opgedeeld in annotatie-eenheden, die niet noodzakelijkerwijze overeenkomen met

**2** Meer informatie over het project en over de distributie van het materiaal via de website <http://www.lands.let.kun.nl/cgn>.

**3** Er blijft natuurlijk altijd wat te wensen over: voorlopig blijven varianten van het Nederlands als gesproken door kinderen, niet-moedertaalsprekers, inwoners van Suriname en de Antillen enzovoorts buiten beschouwing. Niets verbiedt ons echter om dat soort materiaal te gelegener tijd aan het corpus toe te voegen.

**4** ELRA staat voor European Language Resources Association; informatie via <http://www.icp.inpg.fr/ELRA/>.

**5** De vakterm binnen de computationele taalkunde is Part Of Speech tagging of POS-tagging, maar in dit artikel zullen we zo veel mogelijk trachten de klassieke Nederlandse terminologie te hanteren.

**6** In de prosodische annotatie worden de belangrijkste grenzen van woordgroepen (frasegrenzen) alsmede één of twee belangrijkste woorden (zinsaccenten) van elke frase aangeduid.

**7** Behalve punten, vraagtekens en beletseltekens (…) wordt er tijdens de orthografische transcriptie van het corpusmateriaal geen interpunctie aangebracht, omdat het onmogelijk blijkt daarin voldoende consistentie tussen transcribenten te bereiken.

het klassieke begrip 'zin', maar die we toch met die term zullen aanduiden.[8] Een realistisch voorbeeld van zo'n zin is gegeven in (1).[9]

(1) Ik zal u gaan uitleggen hoe we dat zo'n beetje hebben aangepakt dat probleem.

Deze transcriptie van het spraaksignaal is verrijkt met lemma's – dat wil zeggen dat ieder woord is gekoppeld aan een basisvorm in het CGN-lexicon – en de al genoemde taalkundige ontleding:[10]

(2) < au id=1 t=0.000 sp=N00052 >

| | | |
|---|---|---|
| ik | VNW(pers,pron,nomin,vol,1,ev) | ik |
| zal | WW(pv,tgw,ev) | zullen |
| u | VNW(pers,pron,nomin,vol,2b,getal) | u |
| gaan | WW(inf,vrij,zonder) | gaan |
| uitleggen | WW(inf,vrij,zonder) | uitleggen |
| Hoe | BW() | hoe |
| we | VNW(pers,pron,nomin,red,1,mv) | we |
| dat | VNW(aanw,pron,stan,vol,3o,ev) | dat |
| zo'n | VNW(aanw,det,stan,prenom,zonder,agr) | zo'n |
| beetje | N(soort,ev,basis,onz,stan) | beetje |
| hebben | WW(pv,tgw,mv) | hebben |
| aangepakt | WW(vd,vrij,zonder) | aanpakken |
| dat | VNW(aanw,det,stan,prenom,zonder,evon) | dat |
| probleem | N(soort,ev,basis,onz,stan) | probleem |
| . | LET() | . |

Toelichting: De eerste regel van dit voorbeeld bevat een unieke verwijzing naar de locatie van dit fragment in een spraakbestand. Vervolgens zien we drie kolommen: de eerste geeft het getranscribeerde spraaksignaal, een woord per regel, de tweede kolom biedt woordsoorteninformatie (hoofdwoordsoorten in hoofdletters, kenmerken tussen haakjes), en de derde kolom bevat de lemma's, dus de lexicale basisvormen. Bijvoorbeeld, het eerste woord, *ik*, is een voornaamwoord, en wel een volle vorm (in tegenstelling tot geredu-

**8** Miller en Weinert (1998: 30-31) verdedigen zelfs de stelling dat spontane gesproken taal überhaupt niet of nauwelijks zinnen kent die overeenkomen met schrijftaalzinnen, eenheden die beginnen met een hoofdletter en eindigen met een punt. In navolging van Halliday nemen zij aan dat "the language system must be analyzed as having clauses combining into clause complexes" (p. 31). Toch zullen we de term 'zin' gebruiken.

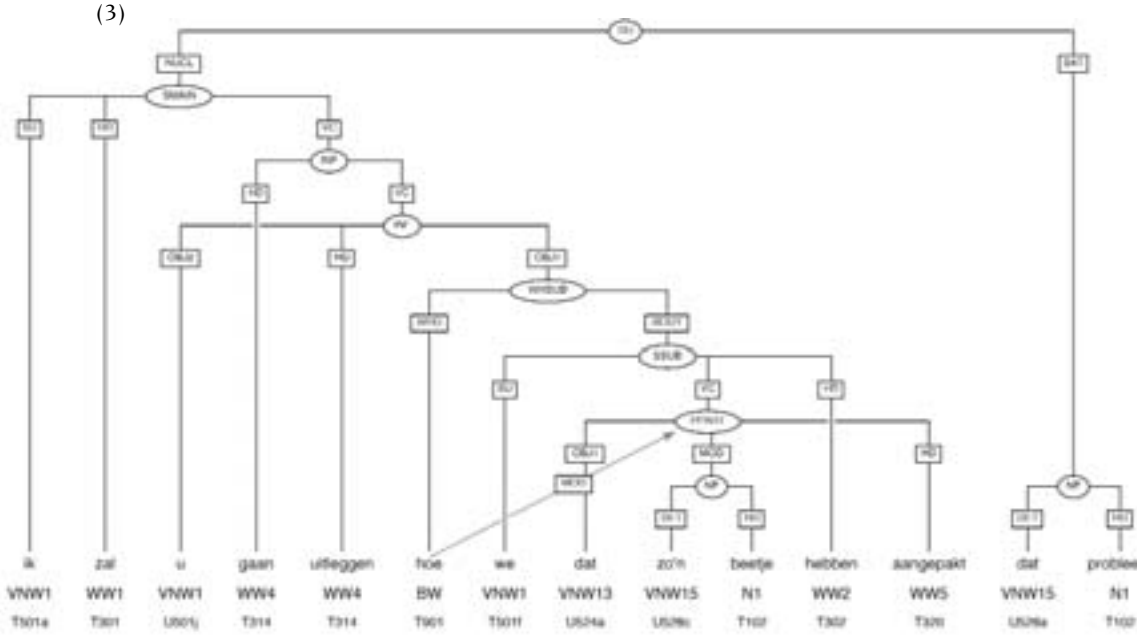
**9** Wegens ruimtegebrek hebben we een relatief korte zin gekozen: vele van de zinnen in het corpus bestaan weliswaar uit een enkel woord, maar er zijn er ook van meer dan 150 woorden. De gekozen zin is bovendien naar verhouding tamelijk welgevormd, maar later in dit artikel komen nog voorbeelden met versprekingen, aarzelingen en dergelijke aan de orde.

**10** De taalkundige ontleding geschiedt op een manier die vergelijkbaar is met de redekundige ontleding, namelijk half-automatisch: het resultaat van een team van automatische woordsoorten-toekenningsprogramma's, die gebruik maken van zo'n 360 woordsoorten-labels en gezamenlijk een precisie bereiken van zo'n 95%, wordt met de hand gecontroleerd en zo nodig gecorrigeerd. Voor details aangaande POS-tagging en lemmatizering binnen het CGN verwijzen we naar Van Eynde (2001) en Van Eynde et al. (2001).

ceerde vormen als *we* op regel 8[11]) van een persoonlijk voornaamwoord van de eerste persoon enkelvoud in de eerste naamval, waarvan de basisvorm *ik* luidt. Het tweede woord, *zal*, is een persoonsvorm, tegenwoordige tijd enkelvoud, van het werkwoord *zullen*, enzovoorts.

(3) geeft een beeld van het soort analyse dat van deze zin voor het CGN wordt gegenereerd (hoe dat gebeurt, komt later aan de orde).[12]

(3)




Het voorbeeld in (3) illustreert een aantal opvallende kenmerken van de CGN-annotatie:

- De annotatie geeft een soort dependentiestructuur (afhankelijkheidsstructuur) en geen constituentenstructuur of functionele structuur. Het resulterende object is dan ook een graaf, en geen klassieke boomstructuur: in zo'n graaf kunnen takken elkaar kruisen, en kunnen dochters meer dan één moeder hebben.

**11** De transcriptie laat ook gereduceerde vormen als *'k* en *'ns* toe, wat laat zien dat de term 'orthografisch' binnen het CGN niet mag worden gelijkgesteld aan 'volgens de officiële spellingregels'. Dialectvormen zoals *benne*(*n*) ('zijn') en *effekes* ('eventjes') worden apart gemarkeerd – vergelijk noot 26.

**12** De onderste rij labels (T501a enz.) is (zonder informatieverlies) afgeleid van de taalkundige ontleding (POS-tags) (vergelijk (2)): "T501a" is bijvoorbeeld een afkorting voor "VNW(pers,pron,nomin,vol,1,ev)". De rij labels daarboven is daarvan afgeleid en vormt een reductie van de oorspronkelijke verzameling, die voor de automatische ontleder onhandig groot is. "VNW1" staat voor 'persoonlijk voornaamwoord', "WW1" voor 'werkwoord (persoonsvorm)', "WW4" voor 'werkwoord (onbepaalde wijs)', enz. De namen van de takken in de graaf staan in rechthoekjes: "SU" staat voor 'onderwerp', "HD" voor 'hoofd', "OBJ2" voor 'secundair object', "NUCL" voor 'kern (van een discourse-eenheid DU)', "VC" voor 'verbaal complement', "OBJ1" voor 'primair object', enz.

- De afhankelijkheidsrelaties zijn onafhankelijk van de oppervlaktevolgorde en de opbouw van de woordgroepen: het werkwoord *uitleggen* bij voorbeeld selecteert een lijdend voorwerp (in dit geval een afhankelijke vraag) aangeduid met OBJ1, en een meewerkend voorwerp *u*, aangeduid met OBJ2. In de oppervlaktevolgorde staat *u* echter tussen de persoonsvorm *zal* van de hoofdzin (aangeduid met HD (hoofd)) en het hulpwerkwoord *gaan* dat het hoofd is van een hoger werkwoordelijk complement (VC). Aldus ontstaan kruisende afhankelijkheden.

- Het vraagwoord *hoe* vervult de rol van hoofd van de afhankelijke vraag WHSUB, maar tegelijkertijd fungeert het als modificeerder binnen de deelwoordgroep PPART, die zelf weer is ingebed in die afhankelijke vraag. Deze dubbele functie wordt uitgedrukt door de twee dependentielabels WHD (hoofd van een vraagwoordconstructie) en MOD, die *hoe* met de twee moederknopen verbinden.[13]

- Elementen in de uitloop (vergelijk Haeseryn et al. 1997: 1397 vv.) – in andere kaders spreekt men wel van "rechtsdislocatie") worden niet beschouwd als onderdeel van de eigenlijke zinssyntaxis. Het discourse-verband tussen de "hoofdzin" en de "naar rechts verplaatste constituent" (in dit geval de naamwoordgroep *dat probleem*) wordt uitgedrukt door beide constituenten samen te nemen in een DU (voor Discourse Unit) waarin ze respectievelijk de rol van NUCL (kern) en SAT (satelliet) vervullen. Als in een latere fase van het project anaforische relaties ook geannoteerd zullen worden, kan er een verband gelegd worden tussen het cataforische (vooruitwijzende) voornaamwoord *dat* in de kern-component en de satelliet-naamwoordgroep *dat probleem*.

- Leestekens, zoals de (in de orthografische transcriptie ingevoegde) punt op de laatste regel van het voorbeeld in (2), worden in de ontleding buiten beschouwing gelaten.[14]

In de volgende paragrafen gaan we nader in op de uitgangspunten en de implementatie van het ontleedproces.

## 4 Uitgangspunten syntactische annotatie

De syntactische structuren die samen met de andere vormen van verrijking en de CGN-geluidsfiles het Corpus Gesproken Nederlands vormen, worden halfautomatisch afgeleid. Later komen we nog terug op enige details van de praktische implementatie, maar duidelijk zij dat het belangrijkste doel van de onderneming niet de parser (ontleedautomaat) is, maar de verzameling ontlede 'zinnen'. Dat impliceert dus een cruciaal verschil met de thema's van sommige andere artikelen in deze aflevering van *Nederlandse Taalkunde*.[15]

**13** In principe is het zelfs mogelijk dat een element of een constituent meer dan twee rollen vervult.

**14** Waarmee we overigens niet willen beweren dat leestekens vanuit taalkundig oogpunt oninteressant zijn – vergelijk Nunberg (1990).

**15** Uit de bijdrage van Van der Beek et al. elders in dit themanummer blijkt echter dat een ontleed corpus wel degelijk een van de nevendoelen is van de Groningse collega's.

Om annotatie en correctie werkbaar te houden moet de annotatie zo eenvoudig mogelijk zijn. Ook is het zaak zoveel mogelijk gebruikers van dienst te kunnen zijn, zodat adoptie van (één versie van) één theoretisch kader ongewenst is. Anderzijds zijn de CGN-gebruikers gebaat bij een zo rijk mogelijke output. Er is daarom gekozen voor een theorieneutraal primair annotatieniveau in termen van afhankelijkheidsstructuren (vergelijk ook Skut et al. 1997), waarbij in het algemeen nauw aansluiting gezocht wordt bij de traditionele Nederlandse zinsontleding, in casu de ANS (Haeseryn et al. 1997).[16]

### *4.1 Output: Annotatiegrafen*

De taalkundige analyse van het CGN verrijkt het materiaal op twee manieren: met categoriale informatie en met informatie over afhankelijkheden. Een voorbeeld: in (3) heeft de woordgroep *zo'n beetje* de categorie NP (zelfstandignaamwoordgroep) gekregen, en bovendien is aangeduid dat die de functie van MOD (bepaling) vervult in de PPART (voltooiddeelwoordgroep) *dat zo'n beetje hebben aangepakt*. De resulterende structuren noemen we afhankelijkheidsstructuren of dependentiestructuren; het zijn, zoals we al zagen, in elk geval geen boomstructuren of constituentenstructuren in de klassieke zin.[17]

### *4.2 Formeel*

Formeel is een CGN-dependentiestructuur D = ⟨K,T⟩ een gelabelde gerichte, acyclische graaf (DAG). We beschikken over disjuncte verzamelingen CAT en DEP voor de labeling van respectievelijk knopen K en takken T.[18]

- Knopen: CAT = POSCAT∪PHCAT: categorielabels (c-labels), de vereniging van lexicale (part-of-speech) en frasale labels ((woord)groeplabels).
- Takken: DEP: dependentielabels (d-labels).

We onderscheiden *gelede* en *ongelede* dependentiestructuren. Een ongelede dependentiestructuur is een knoop met een c-label uit POSCAT, met andere woorden, een subgraaf die enkel een woord bevat. De elementaire bouwstenen van gelede dependentiestructuren noemen we lokale dependentie-*domeinen*. De moederknoop van een dependentiedomein is gelabeld met een frasaal label uit PHCAT. De dochters hebben c-labels uit CAT. De d-labels voor de moeder-dochtertakken worden gevormd door een *hoofd*, samen met de *complementen* en de *modificeerders* van dat hoofd.[19]

**Hoofd** Het hoofd van een dependentiedomein projecteert het c-label van de moederknoop.

**16** De afhankelijkheidsstructuren van het CGN hebben zich inmiddels de facto ontwikkeld tot de standaard voor de computationele syntactische analyse van het Nederlands: vergelijk bijvoorbeeld Bouma et al. (2001).

**17** Puristen zouden kunnen opmerken dat in een zuivere dependentie-ontleding geen categoriale informatie thuishoort, maar dit ter zijde.

**18** In (3) staan de knooplabels in ovale hokjes en de taklabels in rechthoekige.

**19** In speciale gevallen kan een structuur toch verschillende complementen van hetzelfde type hebben, of zelfs twee hoofden. We komen daarop terug.

**Complementen** Het complementatiepatroon bepaalt de interpretatie van het hoofd in termen van thematische structuur. Een complement-label komt per domein hoogstens één keer voor.[20]

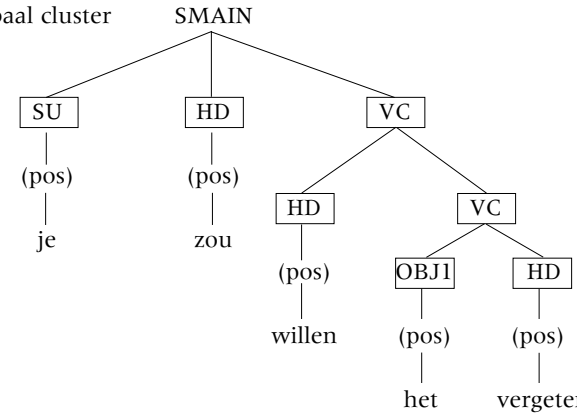
**Modificeerders** Modificerende elementen laten het c-label van de moederknoop ongemoeid; ze kunnen weggelaten worden zonder effect op de thematische structuur. Een en hetzelfde modificeerder-label kan binnen een domein dan ook meerdere keren voorkomen.

*4.3 Consequenties*

Het samennemen van complementatie en modificatie binnen één dependentiedomein leidt tot 'ondiepe' annotatiestructuren. Enkele gevolgen:

- een nieuw domein (hiërarchisch niveau) wordt pas geopend als een nieuw hoofd daar aanleiding toe geeft;
- VOORBEELD: Ondiepe verbale projecties. We onderscheiden *finiete* en *niet-finiete* verbale projecties. De persoonsvorm is hoofd van de finiete zinstypen, de infinitief of het deelwoord van de niet-finiete. Er is dus in de finiete zin geen behoefte aan een tussenliggend VP-niveau.
- complementatie en modificatie zijn *relaties* tussen woordgroepen en een hoofd; als er geen complementen of modificeerders zijn, is er ook geen aanleiding tot niet-vertakkende projecties;
- dependentiedomeinen zijn, in het standaardgeval, *lexicaal verankerd*: het c-label van het hoofd valt samen met de POS-tag;
- VOORBEELD: In de CGN-annotatie hebben werkwoordsgroepen op het dependentieniveau een geneste structuur, gemotiveerd door de onderscheiden subcategorisatie-eisen van de samenstellende hoofden. Bij wijze van voorbeeld: de zin *je zou het willen vergeten* krijgt de volgende dependentiestructuur (we zien even af van de oppervlakte-woordvolgorde):

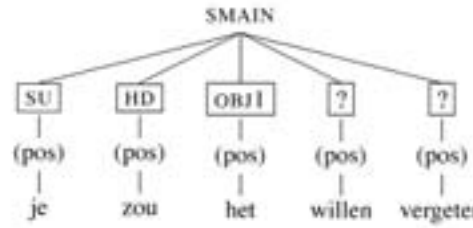
(4) Mogelijke uitvoer: verbaal cluster




**20** Maar vergelijk de vorige noot.

- Op het niveau van (oppervlakte-)constituentenstructuur kan die geneste structuur evenwel ook als een 'platte' reeks werkwoorden worden uitgevoerd, bijvoorbeeld ten behoeve van een HPSG-gebruiker.

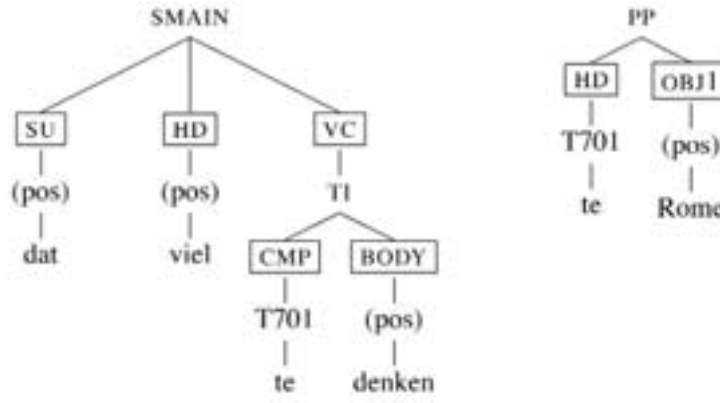
(5) Mogelijke uitvoer: plat




- de eis dat het hoofd het c-label van de moederknoop moet projecteren, houdt in dat we het d-label van het hoofd kunnen laten disambigueren als de POS-informatie voor dat doel niet specifiek genoeg is;

- VOORBEELD: In de woordsoortentoekenning wordt geen onderscheid gemaakt tussen *te* als hoofd van een voorzetselgroep (PP) en *te* als hoofd van een niet-finiete verbale projectie, de *te*-infinitief (TI): beide worden benoemd als voorzetsel (T701). We disambigueren door middel van het d-label voor *te*: het hoofd van de PP krijgt het label CMP.[21]

(6)




**21** In de annotatievoorbeelden in dit artikel staan de c-labels in klein kapitaal, de d-labels zijn ingelijst. De d-labels versieren de *takken* van de annotatiegraaf: het zijn dus geen *knopen*. De POS-informatie wordt ongewijzigd van de POS-annotatie overgenomen, en hier niet altijd verder uitgespeld. Dat *te* het label CMP krijgt, net zoals bijvoorbeeld het voegwoord *dat* van finiete ingebedde zinnen, wil overigens niet zeggen dat we menen dat *dat* en *te* precies dezelfde status hebben – die beslissing laten we graag over aan de theoretici.

- de afhankelijkheidsstructuur is onafhankelijk van de oppervlaktevolgorde;

- VOORBEELD: Kruisende en meervoudige afhankelijkheden. We benadrukten al eerder dat de CGN-annotatie grafen oplevert en geen bomen. Grafen met kruisende takken worden gebruikt om afhankelijkheidsrelaties aan te geven die niet stroken met de oppervlaktevolgorde of met de constituentenstructuur. Constituenten kunnen bovendien meer dan één afhankelijkheidsrol krijgen, en dat is de methode die we gebruiken om niet-lokale afhankelijkheden zoals in bijvoeglijke bijzinnen en constituentvragen te representeren.[22]

  - Enerzijds bepalen de elementen die dit soort configuraties introduceren (constituenten met een vragend voornaamwoord of een betrekkelijk voornaamwoord) het c-label van hun moederknoop, dus zijn het hoofden van afhankelijkheidsstructuren.

  - Anderzijds willen we ook in staat zijn aan te geven wat de rol is die deze elementen vervullen in de rest van de zin; het relevante lokale afhankelijkheidsdomein kan immers willekeurig diep ingebed zijn.

  Voorbeelden van kruisende en meervoudige afhankelijkheden hebben we al ontmoet in de graaf van (3).

De dependentie-annotatie valt dus niet te identificeren met een constituentenstructuur – niet met een (klassieke) dieptestructuur, noch met een oppervlaktestructuur. In de filosofie van het CGN is die laatste hoogstens een *afgeleide* van de dependentie-annotatie.

De dependentiestructuur dient, eventueel na koppeling met het CGN-lexicon, wel voldoende informatie te leveren om automatisch een (oppervlakte-) constituentenstructuur als exportformaat af te leiden. Welke vorm die constituentenstructuur dan aanneemt, kan afhangen van de beoogde gebruikersgroep en van de rol die aan een constituentenstructuur wordt toebedeeld in het geheel van de CGN-annotatieniveaus (zie hieronder).

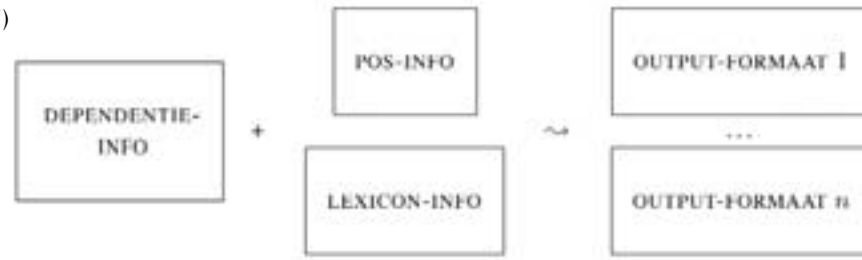
### *4.4 Afgeleide structuren*

Deze primaire annotatiestructuren van het CGN kunnen worden verrijkt met informatie uit de taalkundige ontleding en uit het CGN-lexicon. De combinatie van deze informatiebronnen kan verschillende uitvoerformaten opleveren, die meer of minder toegesneden zijn op de wensen van verschillende gebruikersgroepen.[23]

**22** Specificatie van andere niet-lokale afhankelijkheden, zoals de verwijzing van pronomina en de interpretatie van begrepen subjecten, wordt uitgesteld tot een latere fase van het project.

**23** Details van de syntactische annotatie worden expliciet gemaakt in Moortgat et al. (2002); de laatste versie daarvan wordt steeds meegeleverd op de CGN-CD's.

(7)




Wat betreft afgeleide output-formaten valt te denken aan:

- verrijking van de c-labels met morfosyntactische kenmerken: de verkorte labels kunnen uitgevouwen worden, zodat bijvoorbeeld ook op zulke kenmerken gezocht kan worden;
- verrijking van de d-labels met 'diepe' afhankelijkheden (zoals informatie over semantische controle: begrepen subjecten en dergelijke);
- oppervlakte-constituentenstructuren in een gebruikersvriendelijke notatie (met of zonder 'lege elementen', etc.);
- presentatie-kwesties: keuzemogelijkheden voor de 'taal' van de labels (Nederlands, Engels, ...)

Moortgat en Moot (2001) gaan nader in op de automatische conversie van de syntactische structuren van het CGN. Het zal duidelijk zijn dat sommige vormen van verrijking automatisch kunnen gebeuren, terwijl andere (soms veel) extra werk vragen.

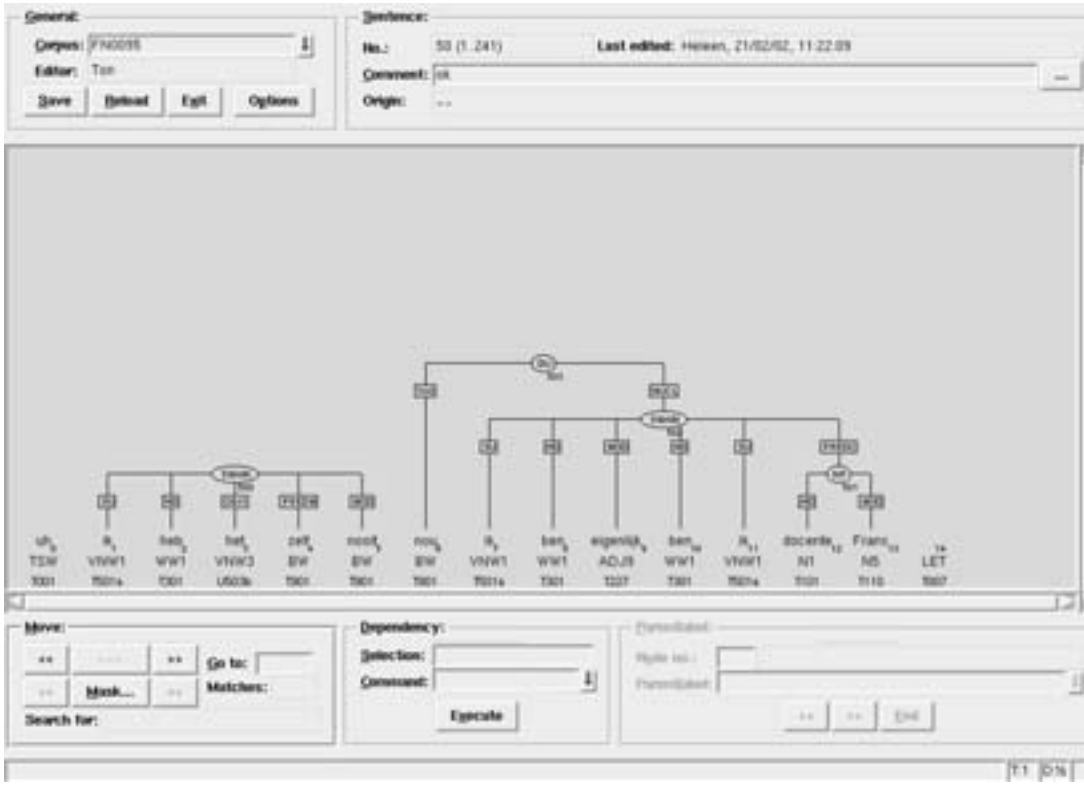
## 5 Methodologie

Onderdeel van de taken van het CGN is dus, als vermeld, de taalkundige ontleding van een (gebalanceerd, gestratificeerd, zo representatief mogelijk) subcorpus van een miljoen woorden. Uitgangspunt voor de ontleding is de orthografische transcriptie, verrijkt met een (grove) indeling in 'annotatie-eenheden' en een fijnmazig systeem van woordsoorten. Met prosodische informatie wordt ook rekening gehouden, maar voorlopig niet in het automatische gedeelte van het proces.

Het proces geschiedt omwille van tijd en consistentie semi-automatisch. We maken gebruik van het programma ANNOTATE, dat in Saarbrücken ontwikkeld is als onderdeel van de zogenaamde NEGRA-tools.[24] De afbeelding in (8) laat zien hoe ANNOTATE zich aan de gebruiker presenteert.

**24** Vergelijk Plaehn (1998) en <http://www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html>.

(8)



ANNOTATE is de buitenkant van een annotatie-omgeving: annotatoren kunnen het programma gebruiken om boompjes, of in ons geval grafen, te construeren voor zinnen. ANNOTATE kan echter ook 'samenwerken' met parsers (ontleedautomaten) van verschillende typen om automatisch, of eventueel na menselijke controle, zinnen te ontleden. Er is voorlopig gekozen voor een ontleder die standaard bij ANNOTATE geleverd wordt, te weten een zelflerend systeem dat ontwikkeld is door Thorsten Brants. Deze ontleedautomaat ontwikkelt op basis van een corpus van ontlede zinnen (in de literatuur staat zoiets bekend als een Tree Bank) en statistiek een theorie over de grammatica die aan die zinnen ten grondslag moet (of zou kunnen) liggen.[25] Die theorie op basis van al ontlede zinnen – in de literatuur bekend als *taalmodel* – kan dan gebruikt worden om het systeem voorstellen te laten doen over mogelijke ontledingen van nieuwe zinnen. Die ontlede zinnen kunnen dan, na controle en eventuele correctie, weer toegevoegd worden aan het corpus, op basis waarvan dan weer een nieuw taalmodel kan worden gegenereerd.

Naarmate er meer werk verzet is, is zo'n statistische grammatica natuurlijk betrouw-

**25** Het onderliggend mechanisme van de ontleedautomaat maakt gebruik van gestapelde Markov-modellen (Cascaded Markov Models of CMM's (Brants 1999)).

baarder – dat is tenminste wat je zou verwachten en wat gerapporteerd wordt voor het Duits. In de praktijk blijkt dat overigens niet eens mee te vallen. Toegegeven, de ontleder heeft nog maar zelden problemen met bijvoorbeeld de correcte ontleding van (eenvoudige) zelfstandignaamwoordgroepen en voorzetselgroepen, maar met hogere structuren gaat het nog steeds teleurstellend vaak mis. Dat dient vermoedelijk ten dele te worden toegeschreven aan het feit dat de ontleder van Brants bedoeld is voor krantentekst, tekstmateriaal dat in principe als welgevormd (dat wil zeggen, in overeenstemming met de regels van de (schrijftaal-)grammatica) kan worden beschouwd en in elk geval veel minder aarzelingen, correcties en versprekingen bevat dan spreektaal. Een gedeelte van die tegenvallende progressie is bovendien waarschijnlijk toe te schrijven aan het gebrek aan homogeniteit van het corpus. We hebben namelijk de indruk dat er grote verschillen zijn tussen, bijvoorbeeld, de interviews met leraren Nederlands, de spontane dialogen en multilogen die bij informanten thuis in de huiselijke kring zijn opgenomen, en de monologen die in de Tweede Kamer zijn opgenomen. Van elk van deze drie teksttypen geven we hieronder een klein fragment.[26]

(9) interview met leerkracht Nederlands

A: uhm lees*a leest u zelf veel uh.
B: mm-hu ja. ja. kranten lezen tijdschriften lezen. ja. ja dat is toch effe voor ‘t ook een beetje voor ‘t werk en ja.

(10) informanten spelen Scrabble

A: oh d’r zijn d’r nog iets van vier of zo.
B: oh d’r zijn d’r nog iets van vier of zo.
A: uhm. oké maar dan zijn d’r dus ook niet erg veel klinkers meer en je hebt nu alleen maar medeklinkers. dus.
B: dan moet ‘k die eerst kwijtraken.

(11) tweedekamerlid in een commissievergadering

de vraag of de ge*a of de minister de garantie wil geven dat de inzet en de aanpak van de sluitende aanpak dat ie niet ten koste gaat van de uh inzet en aandacht voor mensen die al nu al langdurig aan de aan de kant aan ‘t werk*x die garantie is mijn vraag.

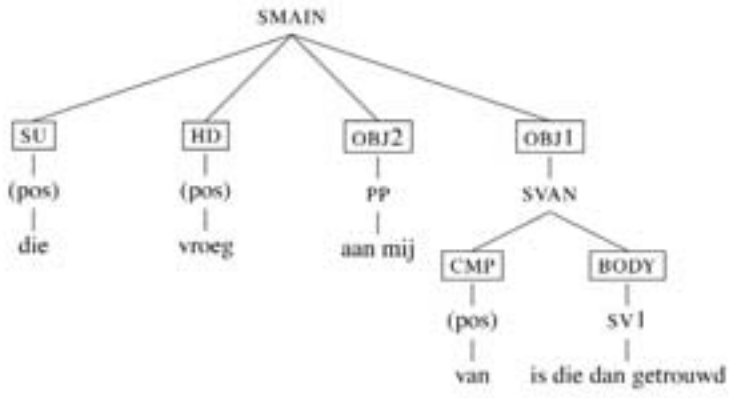
Er bestaan grote taalkundige verschillen tussen de diverse subcorpora van het CGN (Hoekstra et al. 2001, Van der Wouden et al. 2002): de mate van zinsinbedding bijvoorbeeld is in de parlementaire teksten veel groter dan in de interviews met de leraren of de

**26** In de orthografische transcriptie worden de volgende labels gebruikt (Goedertier en Goddijn 2000): *v voor woorden uit een vreemde taal, *d voor dialectwoorden, *z voor dialectisch uitgesproken woorden uit de standaardtaal, *n voor nieuwe woorden (woorden die nog niet in het CGN-lexicon voorkomen), *t voor tussenwerpsels, *a voor afgebroken woorden, *u voor afwijkende uitspraak en versprekingen, *x voor woorden waarvan de transcribeur niet zeker is.

spontane gesprekken. Omgekeerd vinden we in de parlementaire teksten nauwelijks het gebruik van *van* als voegwoord:[27]

(12) Die vroeg aan mij van: is die dan getrouwd?

(13)




Gegeven nu dat er zulke grote verschillen zijn tussen de verschillende subcorpora, is het misschien niet zo verrassend dat het automatisch ontleden niet zo goed gaat. Het ligt immers in de rede dat een statistische ontleedautomaat die getraind is op parlementaire zinnen niet zo heel veel raad weet met een dialoog-zin als (12). Omgekeerd kan training op spontane dialogen er gemakkelijk toe leiden dat bijvoorbeeld een *dat* dat een bijzin inleidt niet als eerste als zodanig zal worden herkend, omdat andere gebruiken van *dat* veel vaker in het trainingscorpus voorkwamen.

## 6 Spreektaalfenomenen

Gesproken taal kent een aantal constructies die in verzorgde geschreven taal in het algemeen als onwelgevormd worden beschouwd en daarom zelden in de literatuur besproken worden (maar vergelijk De Vries (1911), Jansen (1981), De Vries (2001) enzovoorts). Een zo'n fenomeen, het 'expletief' of 'performatief' gebruik van *van*, kwam hierboven al aan de orde in de bespreking van zinnen als (12). Maar er is nog wel meer. Zinsdelen kunnen bijvoorbeeld om discourse- of performance-redenen weggelaten of verdubbeld worden:

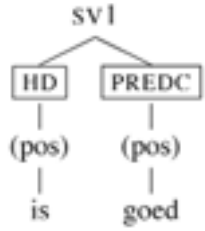
(14) a. Is goed! ('topic-drop': zin zonder onderwerp)
b. Doen we ('topic-drop': zin zonder lijdend voorwerp)

**27** Het is ook mogelijk om dit *van* – dat volgens Van den Toorn et al. (1997: 529) al in de negentiende eeuw voorkwam – op te vatten als de hoorbare tegenhanger van de dubbele punt (Romijn 1999), maar nogmaals (cf. noot 21), dat is niet aan ons om te beslissen.

c. Zijn vader beschuldigt hem dat hij zijn moeder vermoord heeft (zin zonder voorlopig voorzetselvoorwerp)[28]
d. Ik ben eigenlijk ben ik docente Frans ('spiegelconstructie')[29]

Het is niet aan het CGN om te bepalen wat correct Nederlands is en wat niet. Alles wat in het corpus opgenomen is, wordt in principe als grammaticaal beschouwd en moet dus een ontleding krijgen. De ongewone zinstypen in (14) worden dan ook 'gewoon' ontleed: als het zo uitkomt, dan maar zonder onderwerp of lijdend voorwerp, of juist met twee onderwerpen, verbale hoofden enzovoorts. Voor de zin in (14a) wordt dat geïllustreerd in (15), voor de analyse van (14d) verwijzen we naar (8).

(15)




Dit soort spreektaalconstructies dient overigens scherp te worden onderscheiden van een ander spreektaalfenomeen, 'performance-errors' en de daarop volgende 'reparaties':

(16) a. Bij een huwelijk was het vroeger gemakkelijk gezegd: tot de dood hon*u ons scheidt hè.
b. En het is waarschijnlijk het uh misschien het eten van sushi of ik weet niet wat

In (16a) zegt de spreker *hon* waar hij of zij kennelijk *on*(*s*) bedoelt; de fout wordt in elk geval onmiddellijk hersteld. In (16b) begint hij of zij met *het is waarschijnlijk*, bedenkt vervolgens dat dat tot een tè sterke uitspraak gaat voeren, en vervangt het bijwoord door een iets zwakker woord, te weten *misschien*. In deze en gelijksoortige gevallen nemen we aan dat de zinsfragmenten die gecorrigeerd worden daarmee automatisch niet tot de zin behoren; in elk geval worden ze in de syntactische annotatie ook buiten beschouwing gelaten.[30]

Wordt een verspreking niet gecorrigeerd, dan wordt het 'foute' woord gewoon in de annotatie betrokken: in zin (17a) fungeert *Maal* als zelfstandig naamwoord, en *impliciepe*

**28** Een standaardvoorbeeld van dit type constructie is *bananen ben ik dol op*, reden waarom Van der Horst en Van der Horst (1999: 267-72) haar aanduiden met de term 'bananenzin'. Zij gaan ook in op de geschiedenis van dit soort constructies.

**29** De term 'spiegelconstructie' is van Huesken (2001); Jansen (1981: h. 8) spreekt in navolging van De Vries (1911) van 'herhalingsconstrukties', terwijl de ANS de term 'overloopconstructies' gebruikt (Haeseryn et al. 1997: 1259 vv.).

**30** Een reviewer van *Nederlandse Taalkunde* is bang dat dit soort zelfcorrecties nu niet meer automatisch terug te vinden is, maar die angst lijkt ongegrond: als de zoektaal maar sterk genoeg is (zie hieronder), dan moet het bijvoorbeeld mogelijk zijn om te zoeken naar woorden die wel in de zin (de rij woorden) maar niet in de syntactische boom voorkomen.

in (17b) is kennelijk bedoeld als een bijvoeglijk naamwoord dat *vooronderstellingen* modificeert, en zo is het dat die woorden worden geanalyseerd.[31]

(17) a. maar ik zou d'r een enorm pleidooi voor willen houden om die Linge en die Maal*u en die Rijn een absoluut eigen ge*a uh uh eigen leven te geven.
b. het verhelderen van begrippen het opsporen van je impliciepe*u vooronderstellingen of het aangeven van de redenen die die vooronderstellingen ondersteunen.

## 7 Toepassingen

In deze paragraaf geven we in het kort een paar voorbeelden van het soort vragen dat het corpus kan helpen beantwoorden. Deze lijst voorbeelden is natuurlijk naar believen uit te breiden.

- ik zoek alle zinnen met een vorm van het werkwoord *geven*, opgesplitst naar
  - intransitief gebruik (*Jan geeft* – bij een kaartspelletje bijvoorbeeld);
  - transitief gebruik (*Jan geeft een feestje*);
  - ditransitief gebruik (*Jan geeft de hond een koekje, Jan geeft een koekje aan de hond*);
  - onpersoonlijk gebruik (*het geeft niks*);
  - andere mogelijkheden?
- ik zoek echte voorbeelden van lange Wh-verplaatsing (*met wie zei je dat je dacht dat de kroonprins gaat trouwen?*).
- ik zoek zinnen met inbeddingen in inbeddingen (*Jan zei dat Piet klaagde dat Henk snurkte; ken jij iemand die iets geschreven heeft dat over meervoudige inbedding gaat? de uit de met een rieten dak getooide villa ontvreemde kunstschatten*).
- ik ben geïnteresseerd in het intonatiepatroon van zinnen met een kale NP in de rol van direct object op de eerste plaats (*boterhammen lust ik graag*).
- er wordt beweerd dat modale partikels (woordjes als *maar, eens, even*, cf. Van der Wouden 2002) maar op één plaats in de zin voorkomen (Krivonosov 1963, De Vriendt en Van de Craen 1986). Zijn er zinnen die daartegen pleiten, bijvoorbeeld zinnen van de structuur

**31** Ook pauzevullers als *uh* vallen in onze visie buiten de syntaxis – waarmee niet gezegd wil zijn dat een goedgeplaatst *uh* op het juiste moment geen (pragmatische of andere) functie zou kunnen vervullen.

X [ADVP p1 p2] Y [ADVP p3 p4] Z

met Y niet leeg, en de beide partikelgroepen (aangeduid met ADVP) en Y dochters van dezelfde moederknoop? Een (onacceptabel) geconstrueerd voorbeeld van zo'n zin is:

ga nu [ADVP eerst maar] in die stoel [ADVP eens even] zitten

waarbij het (welgevormde) cluster *eerst maar eens even* doorbroken wordt door een voorzetselconstituent *in die stoel*.

- is er een voorkeursvolgorde voor de complementen bij ditransitieve werkwoorden?
- is /n/-deletie bij nomina gevoelig voor het onderscheid onderwerp-lijdend voorwerp?
- wat voor intonatiepatronen vind je zoal bij de balansschikking?

Het is niet altijd triviaal om een antwoord te krijgen op zulke vragen. Nogmaals, het corpus is nog in opbouw, en de exploratiesoftware, de hulpmiddelen om het corpus daadwerkelijk te raadplegen, is nog niet af, zodat het op dit moment soms behoorlijk lastig kan zijn verschillende informatielagen met elkaar in verband te brengen. Met name de syntactische annotatie is op dit moment nog niet toegankelijk via het exploratieprogramma COREX (Kilpatrick en Hellwig 2002).

## 8 Besluit

In deze bijdrage zijn we ingegaan op doelstellingen, uitgangspunten en praktische details van het onderdeel syntactische annotatie van het Corpus Gesproken Nederlands. We hebben besproken hoe dat corpus tot stand komt, we hebben kort aandacht besteed aan het soort problemen dat men bij het analyseren van echte spreektaal tegenkomt, en we hebben getracht de bruikbaarheid van het corpus te illustreren met voorbeelden van het soort vragen dat taalkundigen, ongeacht hun theoretische voorkeuren, met dit nieuwe instrument kunnen gaan stellen. De praktijk zal ongetwijfeld uitwijzen dat die taalkundigen nog veel creatiever zullen zijn in het gebruik van het corpus dan bij de samenstelling en verrijking ervan bedacht kon worden.

## Bibliografie

**Bouma, G., G. van Noord en R. Malouf (2001).** Alpino: Wide-coverage computational analysis of Dutch. <http://odur.let.rug.nl/alfa/papers/papers/>.

**Brants, T. (1999).** Cascaded Markov Models. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics EACL-99*, Bergen.

**Eynde, Frank Van (2001).** Part of speech tagging en lemmatisering. Technical report, Centrum voor Computerlinguïstiek K.U. Leuven, <http://lands.let.kun.nl/cgn/publicat.htm>.

**Eynde, Frank Van, J. Zavrel en W. Daelemans (2000).** Lemmatisation and morpho-syntactic annotation for the Spoken Dutch Corpus. In *Computational Linguistics in the Netherlands 1999. Selected Papers from the Tenth CLIN Meeting*, red. Paola Monachesi, 53-62. Utrecht: Utrecht University, Utrecht Institute of Linguistics OTS.

**Goedertier, W. en S. Goddijn (2000).** Protocol voor orthografische transcriptie. Interne publicatie CGN-project, beschikbaar via <http://lands.let.kun.nl>.

**Goedertier, W., S. Goddijn en J.-P. Martens (2000).** Orthographic transcription of the Spoken Dutch Corpus. Proceedings LREC 2000.

**Haeseryn, W. en anderen (red.) (1997).** *Algemene Nederlandse Spraakkunst.* Groningen en Deurne: Martinus Nijhoff en Wolters Plantijn. 2e, geheel herz. dr.

**Hoekstra, H., M. Moortgat, B. Renmans, I. Schuurman en T. van der Wouden (2001a.).** On certain syntactic properties of spoken Dutch. Computational Linguistics in the Netherlands, Enschede, November 2001.

**Hoekstra, H., M. Moortgat, I. Schuurman en T. van der Wouden (2001b).** Syntactic Annotation for the Spoken Dutch Corpus Project (CGN). In *Computational Linguistics in the Netherlands 2000* , red. W. Daelemans, K. Sima'an, J. Veenstra en J. Zavrel, 73-87. Amsterdam/New York: Rodopi.

**Horst, J. van der en K. van der Horst (1999).** *Geschiedenis van het Nederlands in de twintigste eeuw.* Den Haag en Antwerpen: Sdu en Standaard.

**Huesken, N. (2001).** Mirrorsentences. Repetition of inflected verb and subject in Spoken Dutch. Doctoraalscriptie Algemene Taalwetenschap Universiteit Utrecht, <http://www.let.uu.nl/~Nicole.Huesken/personal/scriptie/scriptie.pdf>.

**Jansen, F. (1981).** *Syntaktische konstrukties in gesproken taal*. Diss. Leiden.

**Kilpatrick, P. en Birgit Hellwig (2002).** Corpus Gesproken Nederlands (COREX) version 1.4 Manual. CGN-CD, beschikbaar via <http://lands.let.kun.nl>.

**Krivonosov, A. T. (1963).** *Die modalen Partikeln in der deutschen Gegenwartssprache*. Diss. Humboldt Universität Berlin, Göppingen (1977): Kümmerle.

**Miller, J. en Regina W. (1998).** *Spontaneous spoken speech. Syntax and Discourse.* Oxford: Clarendon.

**Moortgat, M. en R. Moot (2001).** CGN to Grail. Extracting a type-logical lexicon from the CGN annotation. In: *Computational Linguistics in the Netherlands 2000*, red. W. Daelemans, K. Sima'an, J. Veenstra en J. Zavrel, 126-143. Amsterdam/New York: Rodopi.

**Moortgat, M., I. Schuurman en T. van der Wouden (2001).** Syntactische annotatie. Internal working document CGN, Utrecht, CGN-CD's en <http://lands.let.kun.nl/cgn/>.

**Nunberg, G. (1990).** *The linguistics of punctuation.* Menlo Park, Calif. [etc.]: Center for the Study of Language and Information. (CSLI lecture notes 18).

**Oostdijk, N. (2000a).** Building a corpus of spoken Dutch. In: *Computational Linguistics in the Netherlands 1999. Selected Papers from the Tenth CLIN Meeting*, red. Paola Monachesi, 147-157. Utrecht: Utrecht University, Utrecht Institute of Linguistics OTS.

**Oostdijk, N. (2000b).** Het Corpus Gesproken Nederlands. *Nederlandse Taalkunde* 5, 280-284.

**Plaehn, O. (1998).** Annotate: Bedienungsanleitung. Document Projekt C3 Nebenläufige Grammatische Verarbeitung. Universität des Saarlandes, FR 8.7 Computerlinguistik.

**Romijn, K. (1999).** Ik schrijf van niet, maar ik zeg van wel. *TABU* 29, 173-178.

**Skut, W., B. Krenn en H. Uzkoreit (1997).** An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing.* Washington, D.C., via <http://arxiv.org/format/cmp-lg/9702004>.

**Toorn, M.C. van den en anderen (red.) (1997).** *Geschiedenis van de Nederlandse taal.* Amsterdam: Amsterdam University Press.

**Vriendt, S. De, en P. Van de Craen (1986).** Over plaatsingsmogelijkheden van schakeringspartikels. *Interdisciplinair Tijdschrift voor Taal- en Tekstwetenschap* 6, 101-116.

**Vries, J. de (2001).** *Onze Nederlandse Spreektaal.* Den Haag: SDU Uitgevers.

**Vries, W. de (1911).** Dysmelie. Opmerkingen over syntaxis (vervolg). Verhandeling behoorende bij het programma van het gymnasium der gemeente Groningen voor het jaar 1911-1912.

**Wouden, T. van der (2002).** Partikels: naar een partikelwoordenboek voor het Nederlands. *Nederlandse Taalkunde* 7, 20-43.

**Wouden, T. van der, H. Hoekstra, M. Moortgat, B. Renmans en I. Schuurman (2002).** Syntactic Analysis in the Spoken Dutch Corpus (CGN). In: *Proceedings of the third International Conference on Language Resources and Evaluation*, red. Manuel González Rodríguez Carmen Paz Suárez Araujo, 768-773. Paris: ELRA.

# Een brede computationele grammatica voor het Nederlands

LEONOOR VAN DER BEEK, GOSSE BOUMA EN GERTJAN VAN NOORD*

**Abstract**

**We present a wide-coverage computational parser and grammar for Dutch. The grammar is developed within the tradition of Head-driven Phrase Structure Grammar and provides detailed and linguistically motivated accounts for most syntactic phenomena in Dutch. A broad overview of the grammar and lexicon is given, as well as a more detailed discussion of the analysis of comparatives and partitive constructions. Parsing and disambiguation uses statistical information, which is derived in part from a syntactically annotated tree-bank. The accuracy and coverage of the grammar is evaluated on representative portions of the tree-bank and a number of problematic constructions for the current grammar are identified.**

## 1 Inleiding

In dit artikel geven we een overzicht van de Alpino-grammatica, een brede computationele grammatica voor het Nederlands die het mogelijk maakt grote hoeveelheden tekst automatisch van een gedetailleerde syntactische analyse te voorzien.[1] De grammatica onderscheidt zich van theoretische grammatica's doordat ze nadrukkelijk is ontworpen met als doel zoveel mogelijk van de syntactische structuren te beschrijven die in vrije tekst worden aangetroffen. Ze onderscheidt zich van de meeste robuuste, *wide-coverage*, computationele grammatica's doordat de grammatica taalkundig gemotiveerd en handmatig ontwikkeld is, en doordat ook minder eenvoudige aspecten van de syntaxis (zoals de analyse van vraagzinnen of van kruisende afhankelijkheden in *verb raising*-constructies) beschreven worden. Door ontwikkeling en evaluatie van de Alpino-grammatica proberen we inzicht te krijgen in de vraag in hoeverre het mogelijk is met een taalkundig gemotiveerde, handgeschreven, grammatica, syntactische analyse uit te voeren die robuust, accuraat, en efficiënt is.

De Alpino-grammatica maakt in ruime mate gebruik van inzichten uit Head-driven Phrase Structure Grammar (Pollard & Sag 1994), in het bijzonder van de variant in Sag (1997), waarin gedetailleerde regels worden gedefinieerd in termen van algemene struc-

* We danken de beoordelaars van een eerdere versie van dit artikel voor hun gedetailleerde commentaar. Adres van de auteurs: Rijksuniversiteit Groningen, afdeling Alfa-informatica, Postbus 716, 9700 AS Groningen. E-mail: vdbeek@let.rug.nl, gosse@let.rug.nl, vannoord@let.rug.nl

1 De Alpino-grammatica is ontwikkeld binnen het NWO Pionier-project *Algorithms for Linguistic Processing*.

turen en principes. In paragraaf 2 introduceren we de theoretische uitgangspunten en het formalisme van de grammatica. In paragraaf 3 geven we een overzicht van het lexicon en de regels die samen de Alpino-grammatica vormen.

Parallel aan het ontwikkelen van de grammatica is begonnen met de opbouw van een syntactisch geannoteerd corpus. Zo'n corpus is om een tweetal redenen van groot belang. Ten eerste wordt de Alpino-grammatica ontwikkeld door meerdere personen, gedurende een periode van tenminste enkele jaren. De omvang en complexiteit van de grammatica maakt dat handmatige evaluatie en controle op fouten moeizaam en arbeidsintensief wordt. Een geannoteerd corpus biedt de mogelijkheid om objectief te bepalen of veranderingen in de grammatica leiden tot een verbetering of niet. Ten tweede kent de grammatica aan iedere zin van enige omvang een groot aantal lezingen toe, een probleem dat met het groeien van de grammatica alleen maar toeneemt. Het kiezen van de juiste analyse is daarom een probleem. De Alpino-grammatica maakt gebruik van statistische modellen om de meest waarschijnlijke analyse van een zin te bepalen. Training en, met name, evaluatie van zulke modellen vereisen een geannoteerd corpus.

Binnen het project Corpus Gesproken Nederlands (CGN) (Oostdijk 2000) zijn richtlijnen ontwikkeld voor de syntactische annotatie van het Nederlands. Deze annotatierichtlijnen worden, op enkele kleine verschillen na, ook gehanteerd bij de opbouw van het Alpino-corpus. In tegenstelling tot het CGN bestaat het Alpino-corpus uit geschreven materiaal (ontleend aan het Eindhoven-corpus (Uit den Boogaart 1975)). In paragraaf 4 bespreken we hoe de Alpino-grammatica kan worden gebruikt als hulpmiddel bij het annotatieproces en hoe statistische desambiguatie verloopt.

In paragraaf 5 presenteren we een methode om de resultaten van syntactische analyse en desambiguatie te evalueren aan de hand van het geannoteerde corpus. We presenteren enkele kwantitatieve resultaten die een indruk geven van de mate waarin automatische syntactische analyse van vrije tekst succesvol is. Naar aanleiding van deze resultaten kunnen we bovendien een overzicht geven van fenomenen in het corpus die voor de huidige versie van de grammatica nog problematisch zijn.

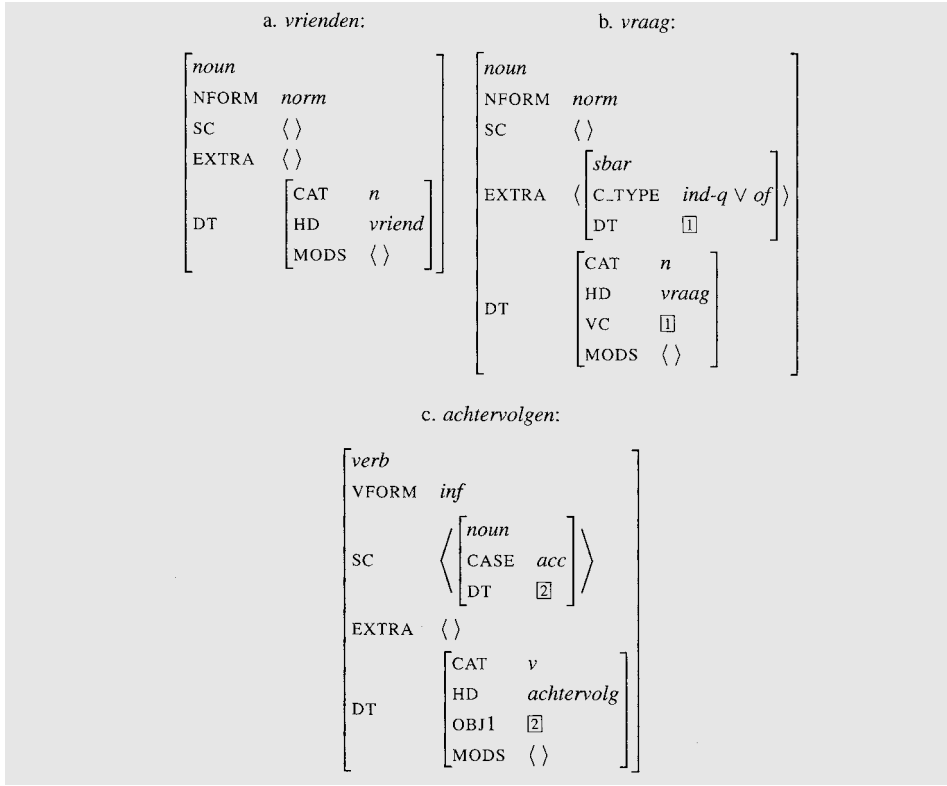
In dit artikel bespreken we de grammatica die door het Alpino systeem wordt gebruikt. We gaan verder niet in op het gebruikte *parseeralgoritme* dat op grond van deze grammatica en een gegeven zin de analyses uitrekent, maar we volstaan met de opmerking dat de Alpino-parser een efficiënte Prolog implementatie van een *left-corner parser* is, zoals uitgebreid beschreven in Van Noord (1997). Deze parser construeert in eerste instantie een *parse forest*: een compacte representatie van alle mogelijke syntactische analyses. Op basis van het desambiguatiemodel probeert het systeem hieruit vervolgens de juiste syntactische analyse te selecteren. Het desambiguatiemodel wordt besproken in paragraaf 4.

## 2 Uitgangspunten

Head-driven Phrase Structure Grammar (Pollard & Sag 1994, Sag & Wasow 1999) is een taalkundige theorie die tracht om formeel precieze beschrijvingen en verklaringen te geven van taalkundige fenomenen. Omdat computationele overwegingen bij het ontwerp van het formalisme niet buiten beschouwing zijn gebleven, is het een theorie die ook binnen de computationele taalkunde aanzien geniet. Brede computationele grammatica's die

gebaseerd zijn op HPSG bestaan onder andere voor het Engels, Duits, en Japans. HPSG-analyses van aspecten van de Nederlandse syntaxis zijn onder andere te vinden in Van Noord & Bouma (1994), Van Eynde (1996, 1999) en Bouma (2000).

HPSG maakt gebruik van *attribute-value matrices* (AVM's) in de definitie van de grammatica. In het woordenboek worden woorden gekoppeld aan een AVM die de lexicale eigenschappen van het betreffende woord representeren. De woorden *vrienden* en *vraag* in figuur 1 zijn van het type *noun*. Hieruit volgt dat bijvoorbeeld het attribuut NFORM is gedefinieerd. De waarde [NFORM *norm*] onderscheidt 'normale' zelfstandige naamwoorden van expletieven als *het* en *er*. Voor het type *verb* is VFORM gedefinieerd. Het attribuut SC bevat een lijst van elementen die als complement bij dit woord kunnen optreden. De infiniefvorm van het werkwoord *achtervolgen* selecteert bijvoorbeeld een direct object.[2] Het zelfstandig naamwoord *vraag* kan combineren met een bijzin (in de vorm van een indirecte vraag of ingeleid door *of*) als complement. Omdat dit complement geextraponeerd kan worden staat het op EXTRA in plaats van op SC. Het attribuut DT representeert een *dependency tree*, een weergave van de grammaticale relaties binnen de constituent waarvan dit woord het hoofd is. De rol van dit attribuut wordt aan het einde van deze paragraaf besproken.




*Figuur 1: Voorbeelden van woorden en de bijbehorende* attribute -value matrices.

**2** De finiete vorm van *achtervolgen* onderscheidt zich onder andere van de niet-finiete vormen doordat SC naast een direct object ook een element bevat dat correspondeert met het subject.

HPSG wordt vaak gepresenteerd als een radicaal lexicalistische theorie, dat wil zeggen, als een theorie die gebruik maakt van een klein aantal algemene regelschema's in combinatie met een rijk gestructureerd lexicon. In Sag (1997) wordt een variant van HPSG voorgesteld waarin ruimte is voor constructiespecifieke regels. Zulke regels zijn vooral nuttig om aspecten van de grammatica te beschrijven die niet gemakkelijk in termen van algemene regels en specifieke lexicale elementen te beschrijven zijn, zoals bijvoorbeeld de syntaxis van relatieve zinnen. Het gevaar dat hierdoor generalisaties worden gemist wordt bezworen door regels te definiëren als instanties van algemene structuren en principes. In de Alpino-grammatica is voor dezelfde aanpak gekozen. Het feit dat parseren op basis van specifieke regels vooralsnog efficiënter is dan parseren met algemene regelschema's is een belangrijk bijkomend voordeel.

Vrijwel alle regels in de Alpino-grammatica zijn instanties van een zogenaamde *headed structure*. Een *headed structure* bestaat uit een moederknoop, een dochter die fungeert als hoofd, en nul of meer andere dochters. Iedere *headed structure* voldoet aan de volgende principes:

- **Head-feature principle**: De attributen die als HEAD features zijn gedefinieerd in de grammatica, worden op de moeder en de *head*-dochter geünificeerd.[3]

- **Valence principle**: De AVM van een eventuele complementdochter dient te unificeren met het eerste element op SC van het hoofd. De SC-waarde van de moeder is de SC-waarde van het hoofd, eventueel minus het element dat correspondeert met een complement dochter.

- **Filler Principle**: De AVM van een eventuele *filler*-dochter dient te unificeren met het eerste element op SLASH van het hoofd. De SLASH-waarde van de moeder is de de SLASH-waarde van het hoofd, eventueel minus het element dat correspondeert met een *filler*-dochter.[4]

- **Extraposition Principle**: De AVM van een eventuele geëxtraponeerde dochter dient te unificeren met het eerste element op EXTRA van het hoofd. De EXTRA-waarde van de moeder is de *concatenatie* van de EXTRA-waardes van alle dochters, eventueel minus het element dat correspondeert met een geëxtraponeerde dochter.

- **Adjunct en Dependency Principle**: De waarde van MODS op de moeder is de concatenatie van de waarde van MODS op het hoofd en de DT-waarde van een eventuele *modifier* dochter. De waarde van alle andere attributen onder DT is identiek op moeder en hoofd.[5]

**3** De unificatie van twee AVM's *A* en *B* is de AVM die precies de informatie bevat die in *A* of *B* aanwezig is. Unificatie mislukt wanneer *A* en *B* tegenstrijdige informatie bevatten.

**4** Zie Bouma et al. (2001) voor meer uitleg en motivatie voor deze *head-driven* benadering van extractie.

**5** Merk op dat adjuncten, in tegenstelling tot de analyse in Van Noord & Bouma (1994) en Bouma, Malouf & Sag (2001), niet in het lexicon geïntroduceerd worden. De lijst MOD onder het attribuut DT dient alleen om de dependentierelatie tussen een adjunct en een hoofd weer te geven, en beregelt niet de selectie van adjuncten.

Het *head feature principle* veronderstelt een onderscheid tussen HEAD features en andere attributen. In standaard HPSG wordt dit onderscheid geïmplementeerd door de HEAD features te groeperen onder een attribuut HEAD. In de Alpino-grammatica worden de HEAD features expliciet opgesomd in de definitie van het *head feature principle*.

Een *head-complement structure* is een specialisatie van de *headed structure*, waarin naast het hoofd precies één complementdochter optreedt. In dat geval volgt uit het *valence principle* dat de waarde van SC op de moeder gelijk is aan de waarde van SC op het hoofd, minus de complementdochter. Aangezien er geen (geëxtraponeerde) *filler*-dochters of adjuncten in de structuur aanwezig zijn, zal de waarde van SLASH en EXTRA op de moeder de concatenatie zijn van deze waarden op de dochters, terwijl de waarde van MODS identiek zal zijn op moeder en hoofddochter.

De *head-filler, head-extra*, en *head-adjunct structures* zijn specialisaties van de *headed structure*, waarin naast het hoofd een dochter optreedt die fungeert als respectievelijk *filler*, geëxtraponeerde dochter, of als adjunct. De waarde van respectievelijk SLASH, EXTRA, of MODS zal in dat geval verschillen op moeder en hoofd, terwijl de waarde van de overige attributen identiek zal zijn.

Verreweg de meeste concrete regels in de grammatica zijn gedefinieerd als instanties van één van bovengenoemde structuren. In de meeste regels dient alleen gespecificeerd te worden wat de categorie en volgorde van de dochters is, en wat het hoofd is. In de voorbeelden hieronder is het hoofd steeds onderstreept:

(1) a. *head-complement-structure: v → np <u>v</u>*
b. *head-adjunct-structure: n → ap <u>n</u>*

Toevoegen van de informatie die uit de principes kan worden afgeleid aan de regels in (1) (in de implementatie gebeurt dit automatisch tijdens het compileren van de regels (*off-line*)), levert de regels op in (2), waarbij de attributen NFORM en VFORM als voorbeeld van een HEAD-feature fungeren. $\langle H|R \rangle$ staat hier voor een lijst met als eerste element $H$ en als staart $R$, $L \oplus M$ representeert de concatenatie van twee lijsten $L$ en $M$.

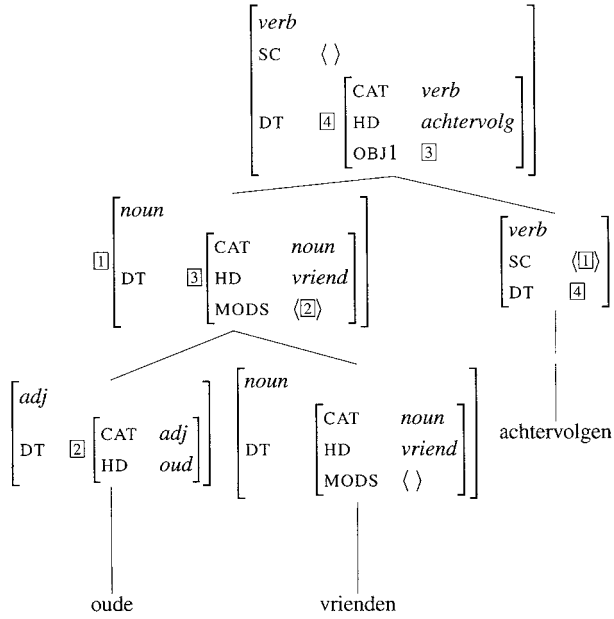
(2) a.
$$\begin{bmatrix} \textit{verb} \\ \text{VFORM} & \boxed{1} \\ \text{SC} & \boxed{2} \\ \text{SLASH} & \boxed{3} \\ \text{EXTRA} & \boxed{4} \oplus \boxed{5} \\ \text{DT} & \boxed{6} \end{bmatrix} \rightarrow \boxed{7} \begin{bmatrix} \textit{noun} \\ \text{EXTRA} & \boxed{4} \end{bmatrix} \quad \begin{bmatrix} \text{VFORM} & \boxed{1} \\ \text{SC} & \langle \boxed{7} | \boxed{2} \rangle \\ \text{SLASH} & \boxed{3} \\ \text{EXTRA} & \boxed{5} \\ \text{DT} & \boxed{6} \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{noun} \\ \text{NFORM} & \boxed{1} \\ \text{SC} & \boxed{2} \\ \text{SLASH} & \boxed{3} \\ \text{EXTRA} & \boxed{4} \oplus \boxed{5} \\ \text{DT} & \begin{bmatrix} \text{CAT} & \boxed{6} \\ \text{HD} & \boxed{7} \\ \text{MODS} & \langle \boxed{9} | \boxed{8} \rangle \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \textit{adj} \\ \text{EXTRA} & \boxed{4} \\ \text{DT} & \boxed{9} \end{bmatrix} \quad \begin{bmatrix} \textit{noun} \\ \text{NFORM} & \boxed{1} \\ \text{SC} & \boxed{2} \\ \text{SLASH} & \boxed{3} \\ \text{EXTRA} & \boxed{5} \\ \text{DT} & \begin{bmatrix} \text{CAT} & \boxed{6} \\ \text{HD} & \boxed{7} \\ \text{MODS} & \boxed{8} \end{bmatrix} \end{bmatrix}$$

De regels maken het mogelijk voor de VP in (3a) de structuur in (3b) af te leiden (waarbij attributen die geen rol spelen zijn weggelaten). De constituent *oude vrienden* wordt afgeleid met behulp van regel (2b) en de VP met behulp van regel (2a).

(3) a. (Kim blijft) oude vrienden achtervolgen

b.




De regels in (4) maken het mogelijk een nominale of een verbale constituent te combineren met een geëxtraponeerde bijzin.

(4) a. *head-extra-structure: n → n sbar*

b. $\begin{bmatrix} noun & \\ AGR & 1 \\ SC & 2 \\ SLASH & 3 \\ EXTRA & 4 \\ DT & 5 \end{bmatrix} \rightarrow \begin{bmatrix} noun & \\ AGR & 1 \\ SC & 2 \\ SLASH & 3 \\ EXTRA & \langle 6 \mid 4 \rangle \\ DT & 5 \end{bmatrix} \; 6\left[sbar\right]$

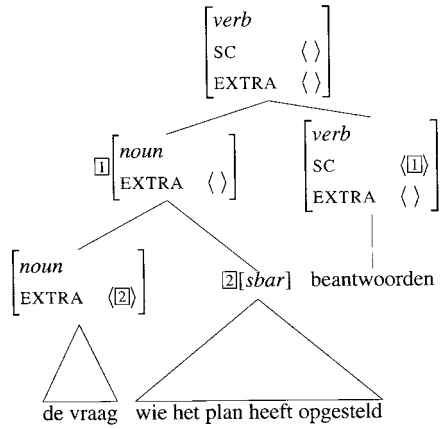
c. *head-extra-structure: v → v sbar*

d. $\begin{bmatrix} verb & \\ AGR & 1 \\ SC & 2 \\ SLASH & 3 \\ EXTRA & 4 \\ DT & 5 \end{bmatrix} \rightarrow \begin{bmatrix} verb & \\ AGR & 1 \\ SC & 2 \\ SLASH & 3 \\ EXTRA & \langle 6 \mid 4 \rangle \\ DT & 5 \end{bmatrix} \; 6\left[sbar\right]$

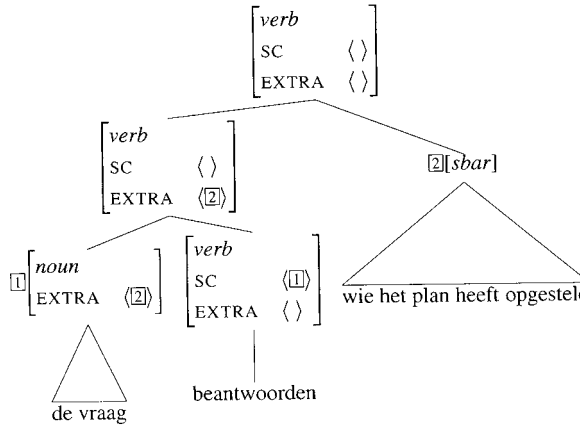
Met behulp van deze regels kunnen de constituenten in (5) worden afgeleid.

(5) a. (Kim moet) de vraag wie het plan heeft opgesteld beantwoorden
b. (Kim moet) de vraag beantwoorden wie het plan heeft opgesteld

c.

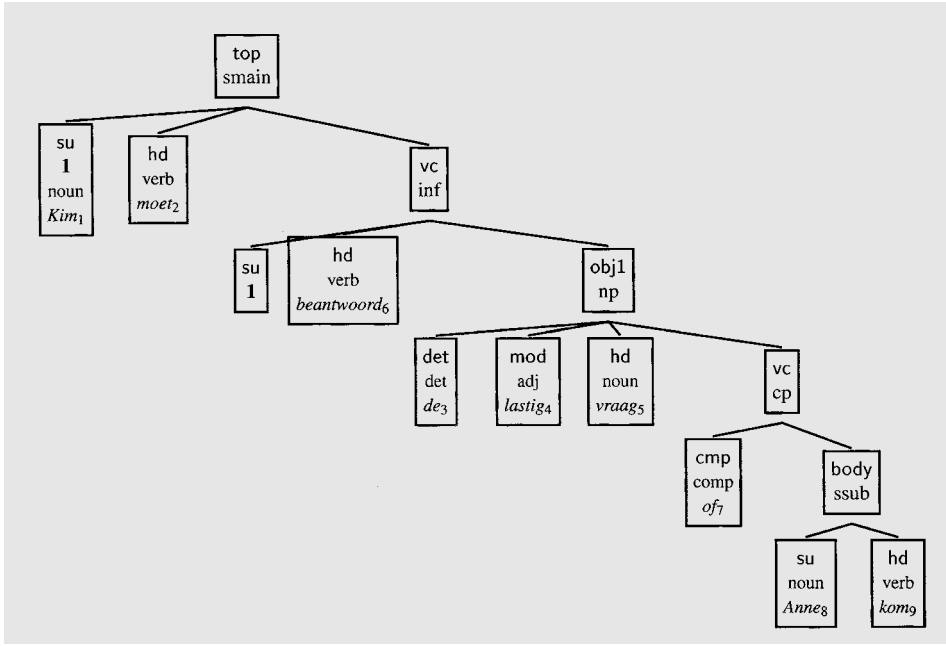



d.




De betekenis van woorden en zinnen wordt in HPSG normaal gesproken weergegeven door semantische representaties aan woorden en regels toe te voegen, samen met principes die de constructie van zulke representaties definiëren. De Alpino-grammatica kent momenteel geen semantische component. Wel worden de grammaticale dependentierelaties gecodeerd, in de vorm van een dependentiestructuur.

Een dependentiestructuur geeft de grammaticale relaties binnen een zin of zinsdeel weer. Een voorbeeld wordt gegeven in figuur 2. De knopen in de boom bestaan uit een grammaticale relatie (*hd* voor *hoofd*, *su* voor *subject*, *mod* voor *modifier*, etc.), eventueel een index (de index **1** geeft aan dat het subject van *moet* identiek is aan het subject van *beantwoord*), en een syntactische categorie. Bladeren bestaan uit een grammaticale relatie, een syntactische categorie, en de stam van een woord met een subscript dat correspondeert met de positie in de zin, of uit een grammaticale relatie en een index. Een belangrijke eigenschap van dependentiestructuur is het feit dat ze abstraheren van woordvolgorde. De




*Figuur 2: Dependency tree voor de zin* Kim moet de lastige vraag beantwoorden of Anne komt.

discontinue string *de vraag ... of Anne komt* kan dus als een eenheid worden weergegeven, ondanks het feit dat het complement is geëxtraponeerd.

Dependentiestructuren worden opgebouwd door aan regels en *lexical entries* een attribuut DT (voor *dependency tree*) toe te voegen. Werkwoorden hebben een DT-waarde waarin is gedefinieerd met welke grammaticale relaties de complementen van het werkwoord corresponderen. De infinitiefvorm van het werkwoord *beantwoorden* selecteert bijvoorbeeld voor een direct object (OBJ1). Dit betekent dat de DT-waarde van dit werkwoord gedefinieerd kan worden als in figuur 1. Uit de DT-waarde voor een zin kan eenvoudig een representatie als in figuur 2 worden afgeleid. De voornaamste redenen om te kiezen voor dependentiestructuren is dat dit een geschikt formaat lijkt om op een theorieneutrale en effectieve wijze een corpus te annoteren met grammaticale structuren (Skut, Krenn & Uszkoreit 1997). Een belangrijk bijkomstig argument is dat op deze wijze kan worden aangesloten bij de annotatierichtlijnen van het Corpus Gesproken Nederlands (Moortgat, Schuurman & Van der Wouden 2000).

## 3 De Alpino-grammatica

De Alpino-grammatica bestaat uit een verzameling regels, gebaseerd op de principes die we hierboven hebben geschetst, en uit een lexicon. Samen beschrijven ze een niet-triviaal deel van de grammatica van het Nederlands. In paragraaf 5 proberen we de dekking van de grammatica preciezer te bepalen.

*3.1Lexicon*

Grammaticale structuren worden in HPSG voor een belangrijk deel bepaald door de lexicale eigenschappen van de woorden, en met name de syntactische hoofden, die deel uitmaken van zo'n structuur. De accuratesse en reikwijdte van de grammatica wordt daarom voor een groot deel mede bepaald door de omvang van het lexicon en de mate van detail waarmee lemma's worden gedefinieerd.

Het Alpino-lexicon bevat momenteel meer dan 70 verschillende verbale valentiepatronen. Een valentiepatroon definieert voor welke complementen, inclusief het subject bij werkwoorden, een hoofd selecteert. De waarde van SC van een woord wordt bepaald door het valentiepatroon voor dit woord. De meeste patronen kunnen ook gebruikt worden met een scheidbaar prefix.[6] Een overzicht van de distributie van verbale valentiepatronen in het lexicon wordt gegeven in tabel 1. Bij de opbouw van het lexicon hebben we gebruik gemaakt van lexicale informatie die voorhanden is in de lexicale databases van Celex (Baayen, Piepenbrock & Van Rijn 1993), Parole,[7] en het CGN (Groot 2000). De Celex-database is vooral nuttig omdat het de verschillende morfologische vormen van een

| Valentiepatroon | Aantal | Voorbeeld |
|---|---|---|
| [SU:NP][OBJ1:NP] | 3438 | zij *aanvaardt* het plan<br>hij *bakent* het plan *af* |
| [SU:NP] | 2158 | zij *aarzelt*<br>hij *barst los* |
| [SU:NP][LD:PP⟨*pform*⟩] | 1389 | zij *arriveert* in Groningen<br>hij *blijft weg* uit Groningen |
| [SU:NP][PC:PP⟨*pform*⟩] | 1271 | zij *ageert* tegen het plan<br>hij *barst* in tranen *uit* |
| [SU:NP][OBJ1:NP][LD:PP⟨*pform*⟩] | 1013 | zij *aait* hem over de bol<br>hij *brengt* de kinderen *onder* bij de buren |
| [SU:NP][OBJ1:NP][PC:PP⟨*pform*⟩] | 855 | zij *achtervolgt* hem met het plan<br>hij *bereidt* haar op het plan *voor* |
| [SU:NP][OBJ1:SDAT] | 418 | zij *aanvaardt* dat het plan mislukt<br>hij *biecht op* dat het plan mislukt. |
| [SU:NP][OBJ2:NP][OBJ1:NP] | 314 | zij *belemmert* hem de doorgang<br>hij *biedt* haar het plan *aan* |
| [SU:SDAT][OBJ1:NP] | 274 | dat het plan kan mislukken *benauwt* haar<br>dat het plan kan mislukken *brengt* onrust *teweeg* |
| [SU:NP][SE:NP][PC:PP⟨*pform*⟩] | 248 | zij *baseert* zich op dit plan<br>hij *geeft* zich *over* aan de politie |

*Tabel 1: De meest frequente verbale valentiepatronen in het Alpino-lexicon. Patronen worden gespecificeerd als een lijst complementen (inclusief het subject), en complementen worden gespecificeerd als grammaticale functie:categorie. De functie* LD *staat voor locatief of directioneel complement,* SE *voor verplicht reflexief complement.*

**6** Scheidbare prefixen worden in de grammatica anders behandeld dan complementen, omdat ze geincorporeerd kunnen zijn in de werkwoordsvorm (*opbelt*), en omdat ze, in tegenstelling tot andere complementen, deel uit kunnen maken van het werkwoordscluster (*heb uit kunnen slapen*).

**7** <http://www.inl.nl/corp/parole.htm>

groot aantal bijvoeglijke en zelfstandige naamwoorden en werkwoorden bevat. Valentiepatronen zijn ontleend aan de lexica van Parole en CGN. Beide lexica specificeren de categorie en grammaticale functie van complementen. Voor het Alpino-lexicon wordt gebruik gemaakt van de vereniging van valentiepatronen uit deze twee bronnen (Bouma 2001). Daarnaast bevat het lexicon valentiepatronen die we zelf hebben toegevoegd. Momenteel zijn dit vooral patronen voor de verschillende hulpwerkwoorden, en voor idiomatische uitdrukkingen zoals *het pleit beslechten*, *als de dood zijn voor iets*, *het eens zijn met iets, op de been blijven, in staat zijn*, etc. Voor bijvoeglijke en zelfstandige naamwoorden zijn een veel kleiner aantal valentiepatronen beschikbaar, voornamelijk voor de selectie van verbale complementen. De betreffende lemma's zijn ontleend aan Parole.

Het Alpino-lexicon bevat zo'n 47.000 lemma's. Het streven is de meest frequente woorden met hun syntactische eigenschappen in het woordenboek op te nemen. Voor het ontleden van vrije tekst (bijvoorbeeld journalistiek proza) betekent dit desalniettemin dat er met een zekere regelmaat woorden voorkomen die niet in het woordenboek staan. Naast eigennamen gaat het hierbij vooral om samenstellingen. De component voor lexicale analyse voorspelt voor woorden die niet in het woordenboek staan een mogelijke categorie op basis van heuristieken die bijvoorbeeld in aanmerking nemen of een woord met een hoofdletter begint, of het woord kan worden geanalyseerd als een samenstelling, etc.

### *3.2 Regels*

De grammatica bevat momenteel ongeveer 330 regels. Bijna de helft van de regels zijn *head-modifier* of *head-complement structures*. De andere helft bestaat uit *head-filler-structures* (voor topicalisatie, WH-vragen, en relatieve zinnen), *head-extra-structures* (voor extrapositie van relatieve zinnen, complementzinnen en VP's, PP's, en comparatiefzinnen), regels voor coördinatie, appositie, verbale constituenten ingeleid door een *complementizer*, en enkele regels voor gesproken taal. De volgende voorbeelden dienen om een globaal overzicht te verkrijgen van de reikwijdte en mate van detail van de regels.

**Complementatie.** De regels in (6) definiëren welke complementen respectievelijk links en rechts van een verbale projectie kunnen voorkomen. Het label *v-arg(left)* staat voor de disjunctie van NP, PP, en AP. Het label *v-arg(right)* staat voor de disjunctie van $\bar{S}$, PP, en VP.

(6) a. *head-complement-structure : v* → *v-arg(left)* $\underline{v}$
b. een boek *kopen*, in Sinterklaas *geloven*, aardig *vinden*
c. *head-complement-structure : v* → $\underline{v}$ *v-arg(right)*
d. *geloven* dat Sinterklaas bestaat, *geloven* in Sinterklaas, *proberen* om te komen

De regels in (7) definiëren de mogelijke complementen van preposities, waarbij *p-arg* staat voor de disjunctie van NP[NFORM *norm*], PP, AP, $\bar{S}$, en VP. Regel (7c) is nodig voor gevallen waar naast een prepositie een partikel aanwezig is en (7e) voor zogenaamde *postposities*.

(7) a. *head-complement-structure : p* → $\underline{p}$ *p-arg*
b. *in* Groningen, *tot* aan de rand, *op* rood, *zonder* dat het opvalt, *zonder* te twijfelen
c. *head-complement-structure : p* → $\underline{p}$ *p-arg part*

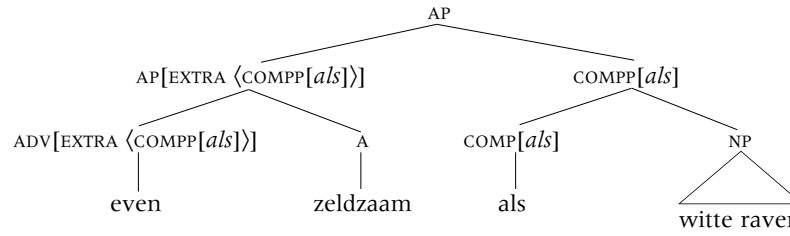
d. *naar* Groningen toe
e. *head-complement-structure : p → p-arg p part*
f. het dak *op*, het bos *in*

**Comparatieven.** Bepalingen van vergelijking (ingeleid door *dan* of *als*) zijn optionele complementen die niet noodzakelijkerwijs direct naast het hoofd staan dat deze bepaling selecteert. Adjectieven in de vergrotende trap (8a)-(8c), de adjectieven (die ook adverbiaal gebruikt kunnen worden) *meer* en *minder* (8d)-(8g), en *(n)iets, (n)iemand), (n)ergens anders, niets/niks* (8h)-(8i) kunnen met een *dan*-bepaling optreden. De combinaties *even* + *adjectief* (8j)-(8l), *(net) zo* + *adjectief* (8m)-(8o), het bijwoord *evenveel* (8p), of een NP met *hetzelfde* of *dezelfde* (8q)-(8s) kunnen optreden met een *als*-bepaling.

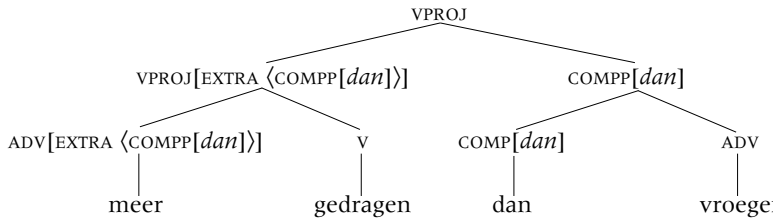
(8) a. Deze prijs ligt *dichter* bij het bod van Bayer *dan* bij dat van Petrofina
b. De internationale fondsen waren aan het slot alle *lager dan* bij opening
c. Zeker nu hij tien kilo *lichter* is *dan* op de dag dat hij bij de Deventerkernploeg werd ingelijfd
d. De gemeenten verkopen *meer* grond *dan* zij kunnen aankopen
e. Leo is *minder* ijverig *dan* zijn broertje
f. Hoeden worden *meer* gedragen *dan* vroeger
g. Een programma waarmee hij zich als artiest *minder* kon afficheren *dan* als vakman
h. *Niks anders* doen *dan* almaar ruw materiaal verzamelen
i. Over van Gaal *niets dan* lof
j. Deze koersen zijn *even* zeldzaam *als* witte raven
k. *Even* duister en ondoorgrondelijk *als* het feit dat het kennelijk het verkeerde resumé was
l. *Even* belangrijk *als* een goed in elkaar getimmerd partijprogramma
m. Tenminste driemaal *zo* groot *als* tien jaar geleden
n. Niet meer *zo* goed *als* vroeger
o. Maar wij zijn *net zo goed* machteloos *als* de regering en de Verenigde Naties
p. Uitgeschakeld worden voor het Jaarbeursstedentoernooi zou *evenveel* betekenen *als* niet meer meetellen in het internationale voetbal
q. Jonge mensen uit *dezelfde* leeftijdscategorie *als* de werkende jongeren
r. Dat was *dezelfde als* gisteren
s. Daar zat *hetzelfde* idee achter *als* bij 't aderlaten

In de grammatica wordt aan lexicale elementen die met een bepaling van vergelijking kunnen optreden, onder andere een definitie toegekend waarin EXTRA een comparatief (COMPP) bevat. Omdat extrapositie is toegestaan op het niveau van AP, NP, en VP, voorspellen we onder meer de mogelijkheid van extrapositie binnen een (predikatieve) adjectivische constituent (9a) en binnen een VP (9b).

(9) a.




b.




De bepaling van vergelijking zelf wordt gevormd door het voegwoord *als* of *dan* gevolgd door een NP, bijwoord, bijzin, VP, A (*witter dan wit*) of PP.

Merk op dat de implementatie van deze analyse gebruik maakt van het feit dat alle regels moeten voldoen aan het *extraposition principle*, en dus een waarde toekennen aan het attribuut EXTRA. Om extrapositie van comparatiefzinnen mogelijk te maken hoeven we dus alleen maar aan te nemen dat adjectieven in de vergrotende trap en een aantal specifieke lexicale items via het attribuut EXTRA selecteren voor een bepaling van vergelijking. Een bijkomend voordeel van de analyse in termen van extrapositie is dat ze de gevallen subsumeert waar de bepaling samen met het hoofd een constituent vormt (*lager dan bij de opening was de koers nog nooit, niets dan lof was er voor van Gaal*), en dat ze het verplichte karakter van extrapositie bij bijwoorden als *even* en *zo* verklaart (**even als witte raven zeldzaam*, **zo als vroeger goed*) omdat extrapositie op het niveau van ADVP immers niet is toegestaan.[8]

**Modificatie van nomina**. In (10) worden enkele regels voor modificatie van nominale constituenten gegeven. Bij modificatie van nomina wordt een onderscheid gemaakt tussen modificatie door een adjectief, PP, of relatieve zin, en appositities. Omdat appositities met de dependentierelatie APP worden gemarkeerd, is voor de betreffende regels een *head-app-structure* gecreëerd. De categorie *app_n* definieert welke nomina als hoofd kunnen

**8** Extrapositie vanuit een getopicaliseerde constituent is in het algemeen niet uitgesloten: *De vraag is gerechtvaardigd waarom de regering niets doet*. Een beoordelaar wees erop dat dit bij comparatieven een minder goed resultaat geeft: **Lager was de koers nog nooit dan bij opening*. Omdat de grammatica geen onderscheid maakt tussen extrapositie van bepalingen van vergelijking en andere vormen van extrapositie, worden dergelijke zinnen momenteel door het systeem geaccepteerd. Hiervoor is ook wel wat te zeggen. Ten eerste blijken dergelijke voorbeelden in corpora wel voor te komen: *Liever benadrukt hij die tegenstellingen dan de bedrieglijke harmonie* en *Nog eerder zal de machtige Mekong droogvallen dan dat de co-premier zijn macht uit handen geeft (Volkskrant 1997)*. Daarnaast worden in de grammatica *zo ADJ . . . dat* constructies ook als comparatieven behandeld (zoals voorgeschreven door CGN); deze constructie laat extrapositie uit topicalisatie heel gemakkelijk toe: *Zo intens lelijk zijn mijn voeten in de loop van een decennium geworden dat ik de mensenmassa's op het strand er in de zomer niet mee wil lastigvallen (Volkskrant 1997)*.

optreden in constructies als *een zak aardappelen*. Het gaat hier in het bijzonder om maat-aanduiders.

(10) a. *head-adjunct-structure : n → n pp*
b. *familie* uit Amsterdam
c. *head-adjunct-structure : n → n rel*
d. *familie* die niemand kent
e. *head-adjunct-structure : np → np post-np-adv*
f. *Beerta* senior/alleen/zelf/ook, *2 februari* aanstaande,
g. *head-adjunct-structure : n → pn n*
h. Chevrolet *programma*
i. *head-app-structure : n → app_n n*
j. een *zak* aardappelen, het *medium* film,
k. *head-app-structure : n → n np*
l. de *familie* Balemans, de *Oostenrijker* Hermann Nitsch, de *hoofdstad* Luxemburg

**Partitieve genitieven**. Een bijzondere vorm van modificatie zijn partitieve genitieven, die bestaan uit een nominale kern gevolgd door een adjectief met genitief *-s*:

(11) a. 't Is me *wat moois*
b. Dat belooft *niet veel goeds*
c. *Wat voor stoms* heb je nu weer uitgehaald?
d. Het is *niets bijzonders*

De nomina die selecteren voor deze constructie (*iets, wat, niets, niks, niet veel, weinig, genoeg, allerlei, meer* of *wat voor*) vormen een kleine, gesloten klasse. In het lexicon zijn deze nomina van het type *iets_n* (12a), dat alleen voor kan komen in de regel voor partitieve genitieven (12b). De regel verlangt daarnaast een adjectief met *-s*-uitgang als adjunct dochter. Deze vorm van het adjectief wordt geconstrueerd tijdens de compilatie van het lexicon. Ze onderscheiden zich van attributieve en predikatieve adjectieven door de waarde *iets* voor het attribuut AFORM (12c).

(12) a. *iets:*
$$\begin{bmatrix} \textit{iets\_n} & \\ \text{SC} & \langle\,\rangle \\ \text{NFORM} & \textit{norm} \end{bmatrix}$$
b. *head-adjunct-structure: np → iets_n iets_adj*
c. *lekkers:*
$$\begin{bmatrix} \textit{adj.} & \\ \text{AFORM} & \textit{iets} \end{bmatrix}$$

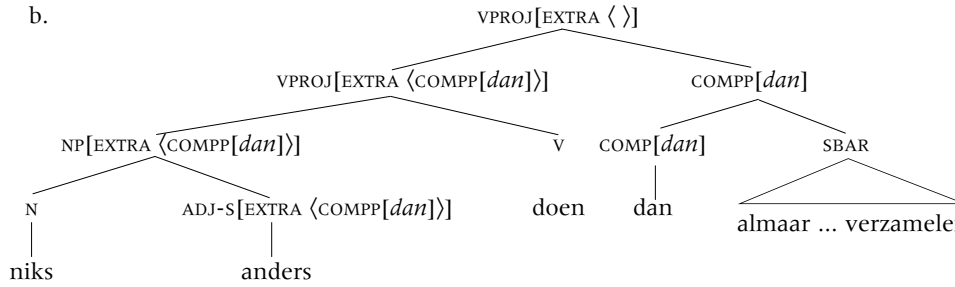
De analyse van de partitieve genitief vereist een aantal tamelijk constructiespecifieke stipulaties, maar maakt toch zo veel mogelijk gebruik van de algemene principes en lexicale categorieën van de grammatica. Doordat de adjectieven in deze constructie zich alleen in het attribuut AFORM van andere adjectieven onderscheiden, kunnen twee genitieve adjectieven bijvoorbeeld gecoördineerd worden met behulp van de algemene regel voor coördinatie van adjectieven:

(13) invallen, waarin ritme en melodie samenvloeiden tot *iets moois maar grilligs*

Een voorbeeld als (14a), tenslotte, demonstreert de interactie van specifieke regels met algemene principes. De NP *niks anders* wordt geanalyseerd als een partitieve genitief. Omdat de regel voor partitieve genitieven een instantie is van een *head-adjunct-structure*, wordt de waarde van EXTRA op de juiste manier doorgegeven, en wordt zonder extra stipulaties voorspeld dat de bepaling van vergelijking die *anders* introduceert vanuit deze constructie geëxtraponeerd kan worden.

(14) a. *Niks anders* doen *dan* almaar ruw materiaal verzamelen

b.




**Overige Regels**. Een beschrijving van de manier waarop enkele lastige aspecten van de Nederlandse syntaxis worden verantwoord, valt buiten het bereik van dit artikel. We volstaan daarom met enkele verwijzingen naar eerder werk. In navolging van Koster (1975) relateren we het finiete werkwoord in hoofdzinnen aan de VP-finale positie waar in bijzinnen het finiete werkwoord te vinden is. De nontransformationele implementatie van dit idee wordt besproken in Van Noord, Bouma, Koeling & Nederhof (1999). Een niet onbelangrijk aspect van de grammatica van het Nederlands is de analyse van verbale eindclusters. Hier maken we gebruik van *argument inheritance* (Hinrichs & Nakazawa 1994). Een verschil met het voorstel in Bouma & Van Noord (1998) is dat in de Alpino-grammatica geen gebruik wordt gemaakt van zogenaamde *linear precedence constraints*, maar dat de woordvolgorde binnen het cluster wordt beschreven door herschrijfregels. De computationele nadelen van *linear precedence constraints* wegen in dit geval zwaarder dan het feit dat hierdoor wellicht enkele generalisaties worden gemist. Constituentvragen en relatieve zinnen maken gebruik van de analyse van extractie zoals voorgesteld in Bouma, Malouf & Sag (2001), met uitzondering van extractie van adjuncten, waarvoor, wederom uit computationele overwegingen, een syntactische in plaats van lexicale analyse is geïmplementeerd.

## 4 Corpusannotatie en Evaluatie

De Alpino-grammatica wordt als hulpmiddel gebruikt bij het construeren van een collectie syntactisch geannoteerde Nederlandse zinnen, de *Alpino Treebank*. De annotatie bestaat, in navolging van het Corpus Gesproken Nederlands (Moorgat et al. 2000), uit

dependentiestructuren. In paragraaf 4.1 wordt kort besproken hoe hierbij te werk wordt gegaan.

Het geannoteerde corpus wordt gebruikt om de kwaliteit van het systeem nauwkeurig te kunnen volgen. Hiertoe wordt het systeem toegepast op de zinnen uit het corpus. De door het systeem als beste beschouwde analyse wordt vervolgens systematisch vergeleken met de geannoteerde versie; deze vergelijking leidt vervolgens tot een kwantitatieve beoordeling van het systeem. Door deze methode kan bijvoorbeeld worden vastgesteld dat toevoegingen aan de grammatica of veranderingen in de grammatica geen onbedoelde problemen veroorzaken. In paragraaf 4.2 laten we zien hoe de *treebank* voor evaluatiedoeleinden wordt ingezet.

Evaluatie aan de hand van de beste analyse die door het systeem wordt geproduceerd veronderstelt een methode om de kwaliteit van analyses te beoordelen. De Alpino Treebank wordt onder andere gebruikt om een statistisch model af te leiden dat in staat is de beste analyse te kiezen uit een verzameling mogelijke analyses. In paragraaf 4.3 geven we een korte schets van de techniek die we hiervoor gebruiken.

### *4.1 De constructie van de Alpino Treebank*

De constructie van de *treebank* behelst het annoteren van een gegeven zin met CGN dependentiestructuren. De annotatie verloopt als volgt. De zin wordt door het systeem automatisch geanalyseerd. Hierbij heeft de annotator de mogelijkheid om in te grijpen in het parseerproces door de lexicale analyse van woorden handmatig te bepalen. Ook kan door middel van het plaatsen van haakjes in de zin de parser in de juiste richting worden gedwongen.

De parser levert in het algemene geval een groot aantal analyses op. Een selectieprogramma, gebaseerd op de SRI TreeBanker (Carter 1997), stelt de annotator in staat snel de juiste analyse te kiezen. Het is hierbij niet nodig de verschillende analyses allemaal één voor één te bekijken. Het komt natuurlijk ook regelmatig voor dat geen enkele analyse de juiste is. In dat geval kiest de annotator een analyse die zo min mogelijk afwijkt van de correcte analyse, en past deze analyse vervolgens handmatig aan met *Thistle* (Calder 2000). Thistle is een programma voor het bewerken en visualiseren van taalkundige structuren. Tenslotte wordt de correct geachte dependentiestructuur als XML-code opgeslagen, en later nog gecontroleerd en mogelijk gecorrigeerd door een tweede annotator.

Momenteel zijn alle 7150 zinnen van het dagbladendeel van het Eindhoven corpus geannoteerd (Uit den Boogaart, 1975). Daarnaast zijn nog enkele kleinere verzamelingen zinnen geannoteerd voor het testen van de grammatica (o.a. de voorbeeldzinnen uit de CGN handleidingen, en voorbeeldzinnen die bij het ontwikkelen van de grammatica zijn gehanteerd).[9]

### *4.2 Evaluatie*

Het syntactisch geannoteerde corpus wordt gebruikt voor evaluatie van de grammatica. Om vast te stellen in hoeverre een door het systeem geproduceerde dependentiestructuur

**9** De huidige versie van de treebank kan worden geraadpleegd op <http://www.let.rug.nl/~vannoord/trees/>

correct is, bepalen we de afhankelijkheidsrelaties die in zo'n structuur aanwezig is. De dependentiestructuur voor de zin in (15a), gegeven in figuur 2, levert de verzameling relaties in (15b) op.

(15) a. Kim moet de lastige vraag beantwoorden of Anne komt

b.

| | | | | | |
|---|---|---|---|---|---|
| ⟨ moet | su | Kim ⟩ | ⟨ moet | vc | beantwoord ⟩ |
| ⟨ beantwoord | su | Kim ⟩ | ⟨ beantwoord | obj1 | vraag ⟩ |
| ⟨ vraag | det | de ⟩ | ⟨ vraag | mod | lastig ⟩ |
| ⟨ vraag | vc | of ⟩ | ⟨ of | body | kom ⟩ |
| ⟨ kom | su | Anne ⟩ | | | |

Relaties bestaan steeds uit drie delen: het woord dat correspondeert met een syntactisch hoofd, de naam van de relatie, en het hoofd van de constituent die in de genoemde relatie tot het eerste woord staat.

Voor evaluatie wordt het aantal relaties van de beste door het systeem opgeleverde analyse ($D_s$) geteld, en van de analyse in de *treebank* ($D_t$). $D_f$ is het aantal foute en ontbrekende relaties in de door het systeem opgeleverde analyse. De *accuratesse* wordt vervolgens gedefinieerd als

$$\text{accuratesse} = 1 - \frac{D_f}{max(D_t, D_s)}$$

Deze formule levert een waarde op tussen 0 (helemaal fout) en 1 (helemaal goed), die grofweg kan worden geïnterpreteerd als het percentage juiste afhankelijkheden.

### *4.3 Desambiguatie*

Evaluatie veronderstelt dat een keuze kan worden gemaakt uit de verschillende analyses van een zin die volgens de grammatica mogelijk zijn.

We maken gebruik van een *log-linear* (*maximum entropy*) statistisch model om een keuze tussen deze analyses te kunnen maken (Johnson, Geman, Canon, Chi & Riezler 1999). Het model telt eigenschappen van een analyse, zogenaamde *features*,[10] die relevant lijken voor desambiguatie. Zo is elke regel in de grammatica een feature, en zijn er features voor de verschillende heuristieken voor het toekennen van taalkundige categorieën aan onbekende woorden. Ook zijn er features die aangeven welke afhankelijkheidsrelaties optreden in een gegeven *dependency structure*. Welke features van belang zijn wordt voorlopig vooral handmatig bepaald (maar zie Mullen (2002)).

Voor een gegeven analyse kunnen we vervolgens bepalen hoe vaak elk mogelijk feature optreedt. In het statistisch model wordt aan elk feature *i* een gewicht $\lambda_i$ toegekend. Een positief gewicht suggereert dat een analyse dat dit feature bevat geprefereerd wordt. Een

**10** De *features* die gebruikt worden voor desambiguatie zijn willekeurige eigenschappen van een syntactische analyse, en vallen dus niet samen met de *features* of attributen die in de grammatica gebruikt worden.

negatief gewicht betekent juist dat het model analyses met zulke features liever niet ziet.[11]

Om een model te construeren moet voor elk feature het corresponderende gewicht worden bepaald. Voor het toekennen van de gewichten gebruiken we een nieuwe implementatie van Maximum Entropy van Rob Malouf (Malouf 2002). De input voor deze berekening bestaat uit een aantal analyses waarbij voor elke analyse de aanwezige features zijn gespecificeerd, en voor elke analyse bovendien wordt aangegeven hoe goed de analyse is. Om deze input te produceren wordt de parser (zonder desambiguatiemodel) toegepast op de zinnen van de *treebank*. De bepaling van de kwaliteit van elke analyse gebeurt door de accuratesse te berekenen, zoals uitgelegd in de vorige paragraaf.

Om voor een gegeven zin de juiste analyse te selecteren moet voor elke analyse de door het model toegekende score worden berekend. Het aantal analyses neemt echter exponentieel toe wanneer de lengte van de zin toeneemt. Figuur 3, waarin het gemiddeld aantal analyses is uitgezet voor zinslengtes tot 20 (volgens de huidige versie van de grammatica), laat zien dat dit fenomeen ook een praktisch probleem oplevert. De parser berekent daarom niet expliciet alle analyses maar construeert voor een gegeven zin een zogenaamd *parse forest*: een compacte datastructuur waarin elke mogelijke analyse makkelijk terug te vinden is. Een zoekprocedure zoekt vervolgens in dit *parse forest* naar de *beste* analyse. De procedure evalueert hiertoe ook gedeeltelijke analyses met het desambiguatiemodel. De procedure vindt heel snel een zeer goede, maar niet gegarandeerd de beste analyse.

## 5 Voorlopige resultaten

Om een indruk te geven van de huidige versie van de grammatica bespreken we hier het resultaat van twee experimenten.

In het eerste experiment wordt de accuratesse van de beste analyse volgens Alpino voor de eerste honderd zinnen van het dagbladendeel van het Eindhoven corpus gemeten. Deze honderd zinnen bevatten gemiddeld zo'n 20 woorden, waarbij achttien zinnen meer dan dertig woorden bevatten. Het desambiguatiemodel dat bij dit experiment wordt gebruikt is getraind op zo'n 3000 andere zinnen uit het Eindhoven corpus. In de linker kolom van tabel 2 is te zien dat slechts in negentien gevallen de door het systeem geprefereerde analyse helemaal correct was. Grote delen van de analyse waren meestal wel in orde: het systeem behaalde een gemiddelde accuratesse per zin van bijna tachtig procent; de totale accuratesse (waarbij de accuratesse wordt berekend over de som van de relaties

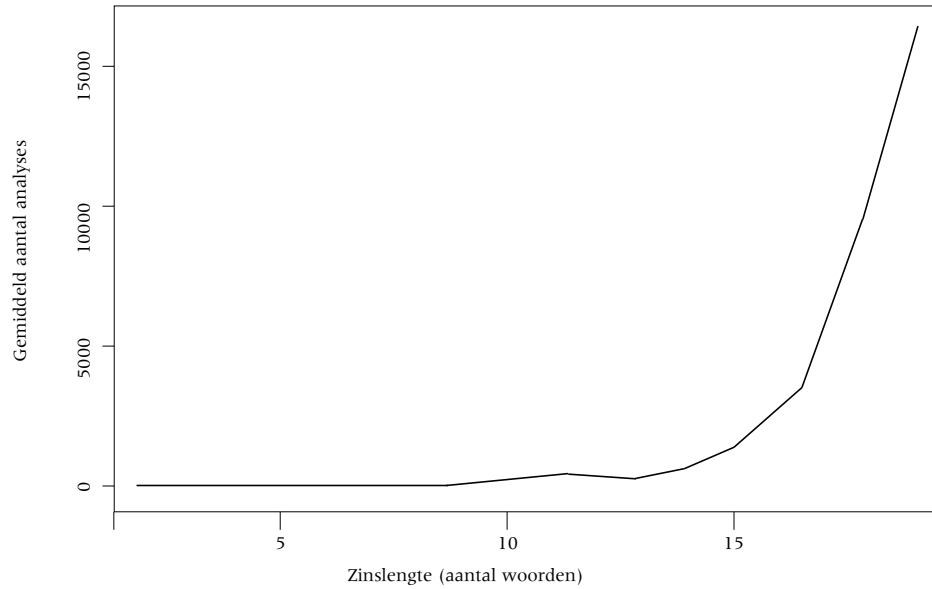
**11.** In zo'n model wordt de kans dat een gegeven zin $x$ de analyse $y$ heeft als volgt gedefinieerd, waarbij $f_i$ staat voor het aantal voorkomens van het feature $i$:

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\sum_i \lambda_i f_i(x, y)\right)$$

De partitiefunctie $Z(x)$ is voor elke parse van $x$ gelijk, en daarom hoeft dit deel van de formule niet te worden berekend: we zijn niet geïnteresseerd in de uiteindelijke waarschijnlijkheid, maar louter in de analyse die de grootste waarschijnlijkheid heeft. Om te bepalen welke analyse de meest waarschijnlijke is hoeven we daarom alleen voor elke analyse de volgende kwantiteit te berekenen:

$$\sum_i \lambda_i f_i(x, y)$$

van alle zinnen samen) was ongeveer 78%. In de gemiddelde accuratesse per zin wegen korte en lange zinnen even zwaar, in de totale accuratesse wegen lange zinnen (die moeilijker zijn) zwaarder dan korte zinnen.




*Figuur 3: Gemiddeld aantal parses per zinslengte*

In het eerste experiment meten we de kwaliteiten van zowel de grammatica als de desambiguatiecomponent. Een interessante vraag is in hoeveel van de gevallen de correcte analyse door de grammatica gemaakt wordt, maar niet als hoogste wordt gekenmerkt door de desambiguatiecomponent. Helaas is dit moeilijk vast te stellen omdat voor de langere zinnen miljoenen analyses mogelijk zijn. Om een indruk te geven van de kwaliteit van de grammatica zonder hierbij de desambiguatie te betrekken hebben we daarom een tweede experiment uitgevoerd. Voor alle geannoteerde zinnen van het Eindhoven corpus met ten hoogste vijftien woorden bleek het wel mogelijk alle analyses uit te rekenen. De resultaten zijn te vinden in de rechter kolom van tabel 2.

| | | |
|---|---|---|
| 100% accuratesse | 19.0 | 77.7 |
| gemiddeld accuratesse per zin | 79.8 | 92.5 |
| totale accuratesse | 78.0 | 94.0 |

*Tabel 2: Accuratesse van de beste analyse volgens Alpino op de eerste honderd zinnen van het Eindhoven corpus (links) en accuratesse van de best mogelijke analyse op 2430 korte zinnen (rechts).*

Uit het tweede experiment blijkt dat, zoals verwacht, desambiguatie een belangrijke invloed op de kwaliteit van het systeem heeft, maar ook is duidelijk dat de grammatica nog voor verbetering vatbaar is. De belangrijkste taalkundige fenomenen uit het corpus die door de grammatica nog niet goed worden behandeld worden hier kort genoemd:

- Ongrammaticale zinnen en spelfouten:

(16) a. Het EEGtoporgaan heeft gisteravond besloten WestDuitsland zijn beperkende maatregelen tegen de import van landbouwprodukten moet opheffen.
b. Ruim dertig percent van de tientallen *mijoenen* Japanse tv-kijkers slaat nooit een aflevering van deze wekelijkse show over.

- Onbekende woorden waarvoor heuristieken niet goed werken:

(17) Langzaamaan werden we bekend.

Het woord *langzaamaan* werd ten onrechte als nomen geanalyseerd.

- Complexe samenstellingen:

(18) a. de 8 procent staatsleningen
b. de 4 x 200 meter ploeg

- Interjecties en andere ingevoegde modificaties (cursief):

(19) a. De Nachtwacht van Rembrandt kun je, *plus hondje*, in levende lijve tegenkomen in Berg en Terblijt.
b. Bij de heren vielen – *maar dat was minder verrassend* – de estafetteformaties volledig door de mand.
c. Ook over de wijk waar ik zelf woon (*Buitenveldert in Amsterdam*) worden door lieden, die dit stadsdeel kennelijk slechts oppervlakkig kennen, de meest krasse veroordelingen uitgesproken.

- Onbekende subcategorisatiepatronen:

(20) Dit jaar ziet men zich al voor problemen gesteld.

- PP-complementen van nomina, extrapositie en topicalisatie van PP-complementen van nomina (in het Alpino lexicon ontbreekt tot nu toe de hiervoor benodigde informatie):

(21) a. Bij de minister werd veel begrip gevonden voor de bij de vakbeweging levende wensen.
b. Op bijna alle brieven hebben we geen reacties ontvangen.

c. Van 't woonhuis bleef een groot gedeelte gespaard.
d. Van dat alles bleef niets heel.

- Elliptische constructies; bepaalde vormen van coördinatie; extrapositie van delen van een coördinatie:

(22) a. Tussen Amsterdam en Schiphol zal de lijn ruim twaalf miljoen reizigers per jaar vervoeren, tussen Schiphol en Leiden ruim elf miljoen
b. De vertegenwoordigers van het gas- en electriciteitsbedrijf zouden vandaag en die van de mijnwerkers overmorgen hun stakingsplannen bekend maken [...]
c. Er worden bloembakken gewenst, goede gordijnen, tafelversiering, wandversiering en sfeervolle verlichting.

## 6 Conclusies

De Alpino-grammatica heeft een veel groter bereik dan eerdere computationele grammatica's voor het Nederlands.[12] Dit is met name te danken aan het feit dat vanaf de start aandacht is besteed aan de prestaties van het systeem op data ontleend aan corpora. Dit heeft ertoe geleid dat er aandacht is besteed aan allerlei vormen van robuustheid, dat er een omvangrijk lexicon is geconstrueerd op basis van algemeen beschikbare elektronische lexica, en dat de grammatica veel gedetailleerder is dan grammatica's die voornamelijk op de taalkundig meest uitdagende constructies zijn gebaseerd.

Veel computationele systemen die geschikt zijn voor het verwerken van grote hoeveelheden tekst beperken zich tot een tamelijk oppervlakkige taalkundige analyse. In de Alpino-grammatica is bewust gekozen voor een formalisme dat alle syntactisch relevante fenomenen kan beschrijven. Zo kunnen bijvoorbeeld constituentvragen, relatieve zinnen, extrapositie, subject- en object-controle, en kruisende afhankelijkheden in werkwoordsclusters op een taalkundig verantwoorde manier behandeld worden. Een juiste beschrijving van de afhankelijkheden die in deze constructies optreden is essentieel wanneer het resultaat van de syntactische analyse moet worden weergegeven als een dependentiestructuur. Een bijkomend voordeel van het gebruik van een formalisme dat gebruik maakt van *attribute-value structures* is het feit dat de constructie van zulke dependentiestructuren eenvoudig in de grammatica zelf kan worden geïncorporeerd.

De constructie van een *treebank* met syntactisch geannoteerd materiaal blijkt inmiddels één van de meest waardevolle nevenproducten van het werk aan de grammatica. We zijn ervan overtuigd dat een dergelijke *treebank*, mits van voldoende omvang, van onschatbare waarde is voor het ontwikkelen van computationele grammatica's en voor het berekenen van statistische desambiguatiemodellen. Bovendien zal een dergelijk corpus een welkom hulpmiddel kunnen zijn voor taalkundig onderzoek, omdat ze het mogelijk maakt de syntactische structuren van het Nederlands in geschreven tekst systematisch te onderzoeken.

**12** Het is moeilijk deze bewering te onderbouwen omdat (voor zover ons bekend is) andere systemen nooit op een vergelijkbare rigoureuze manier zijn geëvalueerd. Bij een recente poging om ontleedsystemen voor het Nederlands te vergelijken (de *"Battle of the Parsers"* tijdens de LOT-winterschool in januari 2001) werd Alpino als winnaar uitgeroepen.

## Bibliografie

**Baayen, R. H., R. Piepenbrock & H. van Rijn (1993).** *The CELEX Lexical Database (CD-ROM).* Philadelphia, PA: Linguistic Data Consortium, University of Pennsylvania.

**Uit den Boogaart, P. C. (1975).** *Woordfrequenties in geschreven en gesproken Nederlands.* Utrecht: Oosthoek, Scheltema & Holkema.

**Bouma, G. (2000).** Argument realization and Dutch R-pronouns: Solving Bech's problem without movement or deletion. In: R. Cann, C. Grover & P. Miller (red.), *Grammatical Interfaces in HPSG,* Stanford, CA: CSLI Publications, 51–76.

**Bouma, G. (2001).** Extracting dependency frames from existing lexical resources. In: *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*. Somerset, NJ: Association for Computational Linguistics, 65–70.

**Bouma, G., R. Malouf & I. Sag (2001).** Satisfying constraints on adjunction and extraction. *Natural Language and Linguistic Theory* 19, 1–65.

**Bouma, G. & G. van Noord (1998).** Word order constraints on verb clusters in German and Dutch. In: E. Hinrichs, T. Nakazawa & A. Kathol, (red.), *Complex Predicates in Nonderivational Syntax,* New York: Academic Press, 43–72.

**Calder, J. (2000).** Thistle and interarbora. In: *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, Saarbrucken, 992–996.

**Carter, D. (1997).** The TreeBanker: A tool for supervised training of parsed corpora. In: *Proceedings of the ACL Workshop on Computational Environments For Grammar Development And Linguistic Engineering*. Somerset, NJ: Association for Computational Linguistics, 9–15.

**Eynde, F. van (1996).** A monostratal treatment of it extraposition without lexical rules. In: W. Daelemans, G. Durieux & S. Gillis (red.), *CLIN 1995, Papers from the sixth CLIN Meeting 1995*. Antwerpen: Universitaire Instelling Antwerpen, 231–248.

**Eynde, F. van (1999).** Major and minor pronouns in Dutch. In: G. Bouma, E. W. Hinrichs, G.-J. M. Kruijff & R. T. Oehrle (red.), *Constraints and Resources in Natural Language Syntax and Semantics*, Stanford, CA: CSLI Publications, 137–152.

**Groot, M. (2000).** *Lexiconopbouw: microstructuur.* Intern rapport van het project Corpus Gesproken Nederlands.

**Hinrichs, E. & T. Nakazawa (1994).** Linearizing AUXs in German verbal complexes. In: J. Nerbonne, K. Netter & C. Pollard (red.), *German in Head-driven Phrase Structure Grammar*, Stanford, CA: CSLI Publications, 11–38.

**Johnson, M., S. Geman, S. Canon, Z. Chi & S. Riezler (1999)**. Estimators for stochastic "unification-based" grammars. In: *Proceedings of the 37th Annual Meeting of the ACL*, Somerset, NJ: Association for Computational Linguistics, 535–541.

**Koster, J. (1975).** Dutch as an SOV language. *Linguistic Analysis* 1, 111–136.

**Malouf, R. (2002).** A comparison of algorithms for maximum entropy parameter estimation. In: *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002).* Taiwan.

**Moortgat, M., I. Schuurman & T. van der Wouden (2000).** CGN syntactische annotatie. Intern rapport van het project *Corpus Gesproken Nederlands*.

**Mullen, T. (2002).** *An Investigation into Compositional Features and Feature Merging for Maximum Entropy-Based Parse Selection*. proefschrift, Rijksuniversiteit Groningen.

**Noord, G. van (1997).** An efficient implementation of the head corner parser. *Computational Linguistics* 23, 425–456.

**Noord, G. van & G. Bouma (1994).** Adjuncts and the processing of lexical rules. In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, Kyoto, 250–256.

**Noord, G. van, G. Bouma, R. Koeling & M.-J. Nederhof (1999).** Robust grammatical analysis for spoken dialogue systems. *Journal of Natural Language Engineering* 5, 45–93.

**Oostdijk, N. (2000).** Het Corpus Gesproken Nederlands. *Nederlandse Taalkunde 5*, 280–284.

**Pollard, C. & I. Sag (1994).** *Head-driven Phrase Structure Grammar*. Stanford, CA: CSLI Publications.

**Sag, I. (1997).** English relative clause constructions. *Journal of Linguistics 33*, 431–484.

**Sag, I. A. & T. Wasow (1999).** *Syntactic Theory: A Formal Introduction.* Stanford, CA: CSLI Publications.

**Skut, W., B. Krenn & H. Uszkoreit (1997).** An annotation scheme for free word order languages. In: *Proceedings of the Fifth Conference on Applied Natural Language Processing*. Somerset, NJ: Association for Computational Linguistics, 88–95.

# ('n) Betekenis berekend

CRIT CREMERS*

**Abstract**

**The computer program Delilah parses complex Dutch sentences and assigns a formal interpretation to them. It does so by applying a particular kind of a combinatory categorial grammar while constructing an underspecified quasi-logical form by unification. This format is spelled out by a post-derivational structure-sensitive algorithm into a set of formulas representing all logical readings. The system lives on detailed lexical information, organized in graphs. The process of unification is driven by specific modes of categorial composition, handling all kinds of constructions. Analyses, prospectives and problems of the grammar exploited by Delilah are discussed.**

## 0 Inleiding

De verhouding – de juiste verhouding – tussen bouw en betekenis is de graal van de moderne taalkunde. Wie weet hoe vormen verwijzen, weet wat taal onderscheidt van alle andere invallen van de evolutie. Het heeft even geduurd, maar langzaamaan lijken de verschillende speurtochten ergens achter de horizon te convergeren. Zo beschrijft Heim en Kratzer (1998) de semantische component van generatieve grammatica nadrukkelijk als de uitvoering van een *Fregean program*, genoemd naar de peetvader van de hedendaagse logische semantiek.

In de ontleedautomaat Delilah kabbelt de strijd om de juiste verhouding op microschaal voort. Het systeem koppelt gedetailleerde syntactische ontledingen van doorwrocht Nederlands aan specificaties van thematische en logische aard. Het wordt aangedreven door een categoriale grammatica die zowel de hiërarchische als de lineaire ordening voor z'n rekening neemt. De ontleder is vanaf 1995 ontwikkeld aan de Universiteit Leiden, in hoofdzaak door Maarten Hijzelendoorn en schrijver dezes.

Deze bijdrage bespreekt enkele grammatische aspecten van het systeem, met name op het raakvlak van vorm- en betekenisanalyse. De grondslagen van de ingebouwde grammatica worden uiteengezet (paragraaf 2) en vervolgens toegelicht aan de hand van relevante verschijnselen (paragraaf 3). Aan het einde komen enkele tekortkomingen van de grammatica en mogelijke remedies aan bod.

* Opleiding Taalwetenschap en ULCL, Universiteit Leiden. Postbus 9515, 2300 RA Leiden, c.l.j.m.cremers@let.leidenuniv.nl. Met veel dank aan Maarten Hijzelendoorn en drie critische anonieme lezers.

## 1 Tussen syntaxis en semantiek

Semantiek gaat over de manier waarop zinnen verwijzen. Omdat zelfs taalkundigen geen idee hebben waar we het 'eigenlijk' over hebben, gaat de semantiek van Delilah over de manier waarop verwijzingsverbanden tussen zinnen worden vastgelegd. Een verwijzingsverband laat zich het best omschrijven in termen van waarheidsvoorwaarden: wat de ene zin waar maakt, maakt een andere ook waar of juist onwaar. Het uitdrukkelijke kader voor het vastleggen en berekenen van dergelijke verbanden is een formele logica. Vandaar dat betekenistoekenning in Delilah de vorm aanneemt van een vertaling naar een logische representatie.

Een belangrijk voordeel van zulke representaties is dat ze allerlei andere soorten propositionele informatie weer kunnen geven, zoals de inhoud van gegevensbestanden. Daarmee zijn ze ook geschikt als invoer en randvoorwaarde voor voortbrenging van natuurlijke taal op semantische grondslag. Delilah voorziet deels in die mogelijkheid, maar hier gaat het alleen over de ontleder. Op korte termijn hopen we zowel de ontleder als de voortbrenger voor toetsing op internet gereed te hebben.

Om een indruk te geven van Delilah's uitvoer, staat hieronder een deel van de automatische ontleding van *enkele zinnen verfden mij groen*. (1) is de afleidingsboom die door inspringing weergeeft welke categorieën met welk resultaat op elkaar zijn toegepast, en hoe de woordgroepen zijn gevormd. (2) bevat delen van het bijbehorende templaat, de eigenlijke ontleding: de gestructureerde benoeming van alle berekende kenmerken van de zin en z'n delen; hierbij zijn de betekenisrepresentaties gecursiveerd; de gebruikte afkortingen spreken wellicht voor zich, maar het gaat om het algehele beeld.

(1)
1-5+s\[]$_-$/[]$_-$ [enkele,zinnen,verfden,mij,groen]
  1-2+np\[]$_+$/[]$_+$ [enkele,zinnen]
    1-1+np\[]$_+$/[n$_0$]$_+$ [enkele]
    2-2+n\[]$_+$/[]$_+$ [zinnen]
  3-5+s\[np$_{wh}$]$_+$/[]$_-$ [verfden,mij,groen]
    3-4+s\[np$_{wh}$]/[ap$_6$]$_-$ [verfden,mij]
      3-3+s\[np$_{wh}$]$_+$/[np$_0$,ap$_6$]$_+$ [verfden]
      4-4+np\[]$_+$/[]$_+$ [mij]
    5-5+ap\[]$_+$/[]$_+$ [groen]

(2)
|ID:1+9
|SYNSEM: |CAT:s
| |CONTROL:controls(experiencer_of(1+9),theme_of(9+14))
| |EXTTH:agent_of(1+9) ...
|LOG:*∃A.(sentence(A)) ∂ (AtPast(9).cause(paint(A, i), green(i)))*
|HEAD: |PHON:verfden
| |SYNSEM: |VTYPE:transacc
| | |FLEX:fin
| | |NUMBER:plur ...

```
|              |LOG:λP.λQ.λV.(AtPast(N).cause(paint(V,Q), P(Q))
|              |TYPE:s\[npwh#(9+2)]+/[np0#(9+11),ap0#(9+14)]+
|ARG(9+11):  |PHON:([],[],[mij],[])
|              |SYNSEM: |OBJ:dirobject_of(1+9)
|              |        |THETA:experiencer_of(1+9)
|              |        |CAT:np    ...
|              |TYPE:np\[]+/0~[]+    ...
|ARG(9+14):  |PHON:([],[],[groen],[])
|              |SYNSEM: |CAT:ap
|              |        |EXTTH:theme_of(9+14)
|              |        |THETA:goal_of(1+9)
|              |LOG:λB.green(B)  ...
|ARG(9+2):   |PHON:([],[],[enkele],[[],[],[zinnen],[]])
|              |LOG:λE.∃A.(sentence(A))  &  E(A))    ...
|TYPE:s\[]_/[]_    ...
|PHON:([[enkele],[[],[],[zinnen],[]]],[],[verfden],[[[],[],[mij],[]],[groen]])
```

Elk element van de ontleding is een paar *KENMERK:waarde*. Zo treft men *FLEX:fin* en *NUMBER:plur* aan, die respectievelijk persoonsvorm-morfologie en meervoud markeren in hun deel van het templaat. Hierbij is *KENMERK* een lid van een eindige verzameling in de grammatica te onderscheiden registers, en *waarde* een vulling van dat register. De waarde kan een constante zijn uit een voor dat register eindige verzameling van mogelijke waardes, een variabele over die mogelijke waardes of zelf een verzameling van paren *KENMERK:waarde*. Voor de logica en systematiek van dergelijke structuren, zie Keller (1993) en Penn (2000).

Het kenmerk *ARG(_)* bevat de ontleding van een zinsdeel dat niet het (syntactische) hoofd van de zin is. Het kenmerk *PHON* omschrijft de vorm van de woordgroep, waarbij aan elke afzonderlijke frase een viertal posities wordt toegekend als aanknopingspunten voor andere frases. *HEAD* geeft steeds de eigenschappen van het syntactische hoofd van een woordgroep. *SYNSEM* is het register voor allerlei syntactische en semantische kenmerken. *TYPE* legt de categorie van de woordgroep vast; de structuur van deze categorieën komt nog uitgebreid aan de orde. Het veld *LOG* bevat de betekenisrepresentatie en wordt hier nader toegelicht.

Elk zinsdeel heeft een templaat van het soort in (2). Zo'n templaat is te beschouwen als een *graaf*: een (vertakkende) structuur van relaties tussen elementen. Het samenvoegen van twee zinsdelen – alle samenvoegingen betreffen tweetallen – tot een geheel komt neer op het *unificeren* van grafen: de structuren worden in elkaar geschoven als ze verenigbaar zijn volgens een bepaald protocol. De verhouding tussen de twee grafen is evenwel asymmetrisch. De ene graaf wordt geünificeerd met een precies omschreven deelgraaf van de ander. De graaf van *enkele* bijvoorbeeld neemt de graaf van *zinnen* in zich op, en de graaf van *verfden* accommodeert de grafen van *enkele zinnen*, *mij* en *groen*.

Elke graaf die bij een unificatie betrokken is, heeft een gespecificeerde waarde voor het kenmerk *LOG* – voor *logische representatie*. De *LOG*-waardes die in (2) staan, zijn het resultaat van speciale uitschrijvingsregels voor de waardes na afloop van de unificatie. Op het moment van unificatie heeft het *LOG*-veld van de 'ontvangende' graaf als waarde een opslag waarin onder meer een positie voor de *LOG*-waarde van de gastgraaf is gemarkeerd

en waarin is vastgelegd hoe deze gastwaarde op kan gaan in een logische vorm voor het geheel. De overdracht van de gastwaarde maakt deel uit van de unificatie. Wanneer dit proces is voltooid, bevat het LOG-veld van de gastheer een *quasi-logische vorm* in de zin van Alshawi (1992). Deze heeft de gedaante van een opslag van *lambda-termen*, d.w.z. van omschreven functies (zie paragraaf 2). In de opslag zijn weliswaar de functie-argument verhoudingen vastgelegd, maar – bijvoorbeeld – nog niet de onderlinge bereiksverhoudingen van de betrokken semantische operatoren. De gedachte dat in ieder geval bereiksgevoelige termen worden opgeslagen, stamt van Cooper (1975); voor een vergelijking met logische methoden in dezen, zie Carpenter (1997:hfdst.7).

Per saldo kent het systeem betekenis dus getrapt toe. Eerst worden de verschillende unificatieopties blootgelegd. Dit lost syntactische ambiguïteit op. Vervolgens krijgt elke eenduidige unificatie een familie van lezingen toegekend. Per unificatie verschillen de lezingen onderling hoogstens in het bereik van operatoren.

## 2 Tussen logica en betekenis

Het unificatieproces in de vorige paragraaf wordt aangestuurd door een *combinatorische categoriale grammatica*. Deze grammatica is beschreven in Cremers (1993). Alhoewel het gaat om een nogal eigenzinnige variëteit van zo'n grammatica, leunt het systeem op de grondslagen omschreven door Steedman (1996, 2000). Categoriale grammatica in algemene zin is terug te voeren op de Poolse logica tussen de wereldoorlogen; zie bijvoorbeeld Ajdukiewicz (1935).

Net als andere categoriale grammatica's, legt de Delilah-grammatica de mogelijkheden tot rangschikking van woordgroepen vast in een beperkt stelsel van samenvoegingsoperaties. Dit stelsel heeft het karakter van een algebra. Dat maakt de rangschikking (de zinsstructuur) berekenbaar. De aard van de algebra leidt ook tot een bepaalde wijze van rekenen en redeneren. De daaruit voortvloeiende afleidingen en bewijzen leveren een patroon volgens welke de woordgroep geïnterpreteerd kan worden. Deze categoriale driehoek van algebra, logica en interpretatie wordt hieronder nader toegelicht. Daarnaast gaat de paragraaf in op de bijzonderheden van de Delilah-grammatica.

### *Categoriale algebra*

Een algebra is een verzameling objecten die gesloten is onder bepaalde operaties: als + zo'n operatie is en *a* en *b* zijn objecten, dan is ook *a*+*b* een object in die verzameling. De leden van zo'n verzameling zijn dan ook te begrijpen als het resultaat van de toepassing van operaties op andere leden, of als een samenstelling uit andere leden, of als herleidbaar op andere leden. Vanwege die operaties is een dergelijke verzameling tenminste gedeeltelijk geordend.

De verzameling van gehele getallen is zo'n algebra, gesloten onder bijvoorbeeld de operaties *optelling* en *vermenigvuldiging*: 17 = 12+5 = (3×4)+(5×1) = ... . De verzameling van rationale getallen (breuken van gehele getallen) is evenzeer een algebra, gesloten onder *vermenigvuldiging* en *deling*. De ordening die in deze algebra's naar voren treedt, is de *kleiner-of-gelijk*-relatie tussen getallen.

Een categoriale algebra definieert en ordent de klasse van welgevormde uitdrukkingen van een taal, zoals een aritmetische algebra een bepaalde klasse van getallen definieert en ordent. Een ordening die hierbij optreedt, is bijvoorbeeld de relatie *is-een-deelreeks-van*. Tegelijk spreekt het vanzelf dat de operatoren van een categoriale algebra anders zijn dan die van een aritmetische. De bijzondere eigenschappen van deze operatoren maken duidelijk hoe de verzameling in kwestie zich verhoudt tot andere systemen met een algebraïsche structuur.

De basisoperatoren van een standaard categoriale grammatica laten zich goed omschrijven als *verketening* (•) en *gerichte deling* (\ en /). Verketening heeft eigenschappen gemeen met aritmetische vermenigvuldiging; gerichte deling lijkt op aritmetische deling. Een lidwoord kan dan gerekend worden tot de klasse van naamwoordgroepen die *rechts gedeeld* zijn door een zelfstandig naamwoord, ofwel: een lidwoord is een representant van de categorie *naamwoordsgroep/naamwoord*, korter: *np/n*. Een aldus getypeerd lidwoord kan aan z'n rechterzijde worden *verketend* met bijvoorbeeld een vertegenwoordiger van de categorie *n*. Het resultaat van deze product-achtige verketening is een uitdrukking van de categorie *np*. In deze procedure wordt de volgende eenvoudige algebraïsche regel toegepast:

(3) $x/y \bullet y \Rightarrow x$

Deze regel is goed te vergelijken met elementaire algebraïsche operaties op getallen, bijvoorbeeld in $3/2 \times 2 = 3$. De analogie tussen verketening en vermenigvuldiging en tussen gerichte deling en deling gaat ver. De categoriale tegenhanger van getalsmatige operaties met vermenigvuldiging en deling is vaak een toegelaten samenvoeging. Vergelijk bijvoorbeeld de 'vermenigvuldiging van breuken' in (5) met de verketening van een voorzetsel – 'neemt een naamwoordsgroep om een voorzetselgroep te maken' – en een lidwoord in (6):

(5) $3/5 \times 5/7 = 3/7$

(6) $pp/np \bullet np/n \Rightarrow pp/n$

Toegepast als reeksvorming, levert dat: als *tussen* een woord (voorzetsel) is van het type *pp/np* en *vele* een (lid)woord van het type *np/n*, dan is *tussen vele* een woordgroep van het type *pp/n*.

Geheel volgens de regels van de rekenkundige algebra, kan nu de reeks *pp/np np/n n* op twee manieren berekend worden, want:

(7) $(pp/np \bullet np/n) \bullet n \Rightarrow pp$; $pp/np \bullet (np/n \bullet n) \Rightarrow pp$

En volgens deze benadering heeft de woordgroep *tussen vele gedichten* ook twee ontledingen:

(8) [ [tussen vele] gedichten]; [tussen [vele gedichten]]

Maar zinnen zijn geen getallen, en dus zijn er grenzen aan de rekenkundige duiding van verketening. De algebraïsche operatoren volgens welke we rekenen, hebben deels gelijke maar deels ook andere eigenschappen dan de algebraïsche operatoren waarmee we

natuurlijke taal verketenen. Enkele eigenschappen van aritmetische vermenigvuldiging en deling zijn hier aangeduid door voorbeelden:

| | | |
|---|---|---|
| (9) | $3\times4 = 4\times3$ | (commutativiteit van vermenigvuldiging) |
| | $3/4 \neq 4/3$ | (non-commutativiteit van deling) |
| | $(3\times4)\times5 = 3\times(4\times5)$ | (associativiteit van vermenigvuldiging) |
| | $(3/4)/5 \neq 3/(4/5)$ | (non-associativiteit van deling) |
| | $(3/4)\times4 = 3$ | (wegstreping) |
| | $3/4 < 3 < 3\times4$ | (ordening) |
| | $3 = 3/1 = 3\times1$ | (eenheid voor deling en vermenigvuldiging) |
| | $3\times0 = 0$ | (nul-element voor vermenigvuldiging) |
| | $3 = 4/(4/3)$ | (verhoging) |

Het gaat hier om eigenschappen van de operatoren omdat de keuze van de getallen, met uitzondering van *1* en *0*, er niet toe doet. Het is nu op z'n minst uitdagend om na te gaan welke verschillen er zijn tussen bijvoorbeeld het stelsel van rationale getallen en een stelsel van verketende woordgroepen, een taal. Dit verschaft ons inzicht in de vraag welk soort algebra taal is.

Stel, een grammatica van een natuurlijke taal is een algebra met twee operatoren: *verketening* ('naast-elkaar-plaatsing') en *gerichte deling* ('groepsvorming'). Wat zijn dan de algebraïsche eigenschappen van deze operatoren, oftewel: hoe verschillen ze van de aritmetische operatoren *vermenigvuldiging* en *deling*? In hoeverre is de ordening die vermenigvuldiging en deling opleggen aan de breuken, te vergelijken met de ordening die verketening en gerichte deling opleggen aan reeksen woorden? Hier zijn enkele observaties:

(10) Verketening is niet commutatief, evenmin als gerichte deling. Immers, het paar *tussen vele* is als 'schakel' in een woordgroep niet gelijk aan *vele tussen,* en een woord van de categorie *pp/np* (een voorzetsel) is niet noodzakelijkerwijs ook van de categorie *np/pp* (naamwoordsgroep met een voorzetselgroep als complement).

(11) Gerichte deling is niet associatief, evenmin als aritmetische deling. Een woord van de categorie *(s/s)/np* – bijvoorbeeld *dat* in de bijwoordelijke bepaling *dat jaar* – is niet ook van de categorie *s/(s/np)* – een typische categorie voor vooropgeplaatste vraagwoorden.

(12) Verketening is wellicht wel associatief. Dat hangt af van de mate waarin we wensen vast te houden aan een vast regime van zinsdelen. Niet iedereen zal de combinatie van een voorzetsel en een lidwoord (bijvoorbeeld *tussen vele* of *vele tussen*) een constituent van een andere woordgroep willen noemen, ook al komen ze naast elkaar in die woordgroep voor.

(13) Verketening en gerichte deling leggen hoogstens een gedeeltelijke ordening op. Woorden zijn onderling nauwelijks geordend en oneindig veel woordgroepen staan niet in enige verketeningsrelatie tot elkaar.

Verder is het duidelijk dat er in taal geen unieke elementen zijn die de rol van 0 of 1 in de getallenalgebra kunnen vervullen. Zelfs de invoering van 'lege categorieën' helpt hier niet, want ook een lege categorie verandert per definitie de reeks. Er is immers geen (taalkun-

dig interessante) algemene stofzuigercategorie *c* te bedenken waarvoor zelfs maar één van de volgende beweringen zinvol zou zijn in de reeksalgebra, voor iedere categorie *a*:

(14) a • c ⇒ a
a • c ⇒ c
a/c ⇒ a

Vanwege de non-commutativiteit van de verketening valt deling in een reeksalgebra uiteen in twee gerichte vormen: deling naar links en deling naar rechts.

(15) a • b\a ⇒ b
b/a • a ⇒ b

Vandaar dat de categoriale algebra drijft op het viertal operatoren • , /, \, en ⇒ waarbij ⇒ de rol speelt van 'zinspredikaat' en gelezen kan worden als: er is een manier om de linkerkant te herleiden op de rechter, of: de categorieën aan de linkerkant verketenen tot de categorie aan de rechterkant.

De verketeningsalgebra verschilt dus op een interessante en niet-triviale manier van andere algebra's. Als belangrijkste verschil met bijvoorbeeld rekenkundige algebra's kan worden aangemerkt dat de samenstelling van een reeks altijd een unieke identificerende eigenschap is. Voor getallen is dat niet zo. Het getal *4* heeft in de rekenkunde talloze schrijfwijzen: *2×2×1*, *7×12/21*, etc. Voor de rekenkundige status van het getal is de wijze waarop het getal geconstrueerd kan worden, van geen belang. Vandaar ook dat we al die constructies als identiek kunnen beschouwen: 7×12/21 = 2×(2/8)×8 = 4 = ... . Een zin of een andere woordgroep van een natuurlijke taal daarentegen wordt volledig bepaald door de wijze waarop ie is samengesteld: twee zinnen zijn pas aan elkaar gelijk als ze op gelijke wijze zijn geconstrueerd en uit dezelfde elementen bestaan.

*Categoriale logica*

Het palet van algebraïsche verschillen en overeenkomsten tussen de constructie van getallenruimte en de constructie van de taalruimte bepaalt het karakter van de categoriale grammatica als vorm van taalkunde. Immers, het verschil in algebra gaat gepaard met een aanzienlijk verschil in de wijze van 'rekenen', d.w.z in de systematiek van het redeneren over het stelsel, dus in de logica. Categoriale grammatica is in wezen het onderzoeken van de wijze waarop met de operatoren *verketening* en *gerichte deling* kan worden geredeneerd. De conclusies van de redering zijn dan beweringen over de welgevormdheid van reeksen op basis van hun categoriale samenstelling. Een stelsel van redeneerstappen heet een rekening of calculus, en omdat de redenering het bewijs is en de conclusie welgevormdheid betreft, is een calculus een grammatica. De stiefmoeder van alle categoriale calculi is de Lambek-calculus (Lambek 1958). Dit systeem legt de verketeningsalgebra vrij dicht tegen de rekenkundige algebra aan, vanwege de volgende redeneerregel:

(16) als je een bewijs hebt voor de verketening van *a* en *b* tot *c*, dus voor de stelling *a* • *b* ⇒ *c*, dan mag je concluderen: *a* ⇒ *c/b* en *b* ⇒ *c\a*

Deze regel komt overeen met de rekenkundige regel:

(17) als (je een bewijs hebt voor) $a \times b = c$ dan (heb je ook een bewijs voor) $a = c/b$ en (voor) $b = c/a$

Aan de toepasbaarheid van deze redeneerstap komt geen einde. Je kunt de stelling daarom ook anders formuleren: elke categorie die combineert met een ander, vertegenwoordigt een eindeloze hoeveelheid andere, meer samengestelde, categorieën. Omdat de verketeningsoperator in de Lambek-calculus ook nog eens associatief is gedefinieerd – net als de rekenkundige vermenigvuldiging – kent een Lambek-grammatica geen vaste constituent-structuur. Er is bewezen (Buszkowski 1988) dat onder de Lambek-rekening geldt: als er een bewijs is voor een bepaalde verketening, dan is er een bewijs ongeacht de groepering van categorieën. Dus:

(18) als er een bewijs is dat *de aap zingt Schubert* (rijtje categorieën bijvoorbeeld: *np/n n (s\np)/np np*) een welgevormde zin is door de combinatie van *de* en *aap* te verketenen met de combinatie van *zingt* en *Schubert* volgens het stramien *[[de aap] [zingt Schubert]]*, dan is er ook een bewijs volgens het stramien *[[de [aap zingt]] Schubert]* en *[de [aap [zingt Schubert]]]*, en omgekeerd

De notie *constituent* is daarom in een Lambek-kader geen primitief begrip.

Logisch gezien komt het toepassen van de Lambek-rekening erop neer dat gerichte deling wordt behandeld als een conditionele bewering: de categorieën *a/b* en *a\b* staan dan voor de stelling *als b dan a*. Het rekenen met categorieën is zo weer te geven als het opwerpen en weer terugtrekken van hypotheses binnen bepaalde beperkingen. Het bijzondere van de Lambek-calculus als voorwaardelijke logica is dat een hypothese (*stel we hebben hier een np; dan ..*) in een redenering (afleiding, bewijs) maar één keer gebruikt of ingetrokken kan worden (*dus: áls we een np hebben, dan...*). Deze eigenschap van de Lambek-logica, uitvoerig besproken in bijvoorbeeld Van Benthem (1991), is kenmerkend voor z'n toepassing bij verketening. Immers, in talige reeksvorming is elk voorkomen van een grootheid apart een gebeurtenis van belang, onomkeerbaar, onuitwisbaar en onherhaalbaar. Dat is niet triviaal. In de 'gewone' propositielogica is bijvoorbeeld de conjunctie *p en p* gelijkwaardig aan *p*. Maar er zijn weinig talen met een grammatica die de reeks van twee elementen *ww* als woordgroep systematisch gelijk stelt aan *w.* Evenmin is het gebruikelijk dat een en dezelfde woordgroep twee verschillende rollen speelt. In de grammatica van natuurlijke taal is elk talig element combinatorisch relevant en combinatorisch bepaald. Grammatica's van natuurlijke talen – verketeningsystemen bij uitstek – zijn dus uitermate gevoelig voor het voorkomen van (talig) materiaal (redeneerstof); ze zijn *resource sensitive*. Daardoor wijkt hun logica op een interessante wijze af van redeneersystemen over andere, bijvoorbeeld propositionele domeinen. Overigens ligt de logica van verketening in dit opzicht weer niet zo ver af van de rekenkundige logica.

In ieder geval levert de logica de grondslag voor een ontleedprocedure, voor een manier van rekenen met taal.

*Categoriale functies*

De redenering volgens welke een reeks herleid kan worden op een bepaalde categorie – dus: de redenering volgens welke een woordgroep een welgevormde uitdrukking is van een bepaalde categorie – levert een afleiding of derivatie van die woordgroep op. De logica van de grammatica legt de toegestane redeneerstappen vast. Een toepassing van opeenvolgende redeneerstappen leidt tot een derivationele structuur: bij elke redeneerstap wordt een operator weggewerkt of toegevoegd, en daarmee wordt de samenhang tussen de deelreeksen bepaald. Deze afleidingen vertegenwoordigen, net als de structuurdiagrammen van de representationele grammatica's, de eigenlijke taalkundige boodschap: de derivatie is de grondslag voor de (taalkundige) interpretatie. Voor dezelfde woordgroep is vaak meer dan één afleiding mogelijk, zoals al eerder is besproken. Het is nog erger: onder de Lambek-rekening laten alle woordgroepen zich op onbeperkt veel manieren afleiden. Deze uitbarsting wordt in de hand gehouden door categorieën te interpreteren als functies die op elkaar worden toegepast volgens de meest algemene functierekening die wij kennen: de *lambda-rekening*. Lambda-termen (korter: λ-termen) kunnen worden ingezet als codes voor de manier waarop categorieën versmelten, onder abstractie van bijvoorbeeld richting. Ze geven daarom de wezenlijke afhankelijkheden weer, eigenlijk: die afhankelijkheden die voor de interpretatie van belang zijn. De λ-termen hebben dus een dubbele missie: ze bepalen klassen van gelijkwaardige afleidingen – afleidingen die op dezelfde λ-term uitkomen – en ze definiëren de logische betekenis van de afgeleide reeks. Elk lexicaal element brengt behalve een categorie een λ-term mee, als betekenisdrager. Deze λ-term definieert een betekenisfunctie die dienst doet als de interpretatie van het woord in kwestie. De functie correspondeert met de combinatorische opties van de categorie: de categoriale algebra kan worden afgebeeld op de funktie-algebra.

Als woordgroepen met elkaar verketenen volgens een algebra en een logica die als functietoepassing en -samenstelling in λ-termen neerslaat, dan verenigen λ-termen in zich derivatie, structuur en betekenis. Dit is de taalkundige winst van categoriale grammatica: de combinatoriek is door de onderliggende logica verbonden met directe toekenning van betekenis. Betekenisvolle uitdrukkingen worden op een betekenisvolle manier tot een betekenisvolle uitdrukking samengesmeed.

De zo verkregen semantische representatie leent zich goed als invoer voor een inferentiële semantiek: die benadering van betekenis waarbij niet de verwijzing maar de semantische consequentie centraal staat, die niet gaat over de vraag: waar gaat dit over? maar over de vraag: als dit zo het geval is, wat volgt daaruit?

Aldus biedt een categoriale benadering van natuurlijke talen samenhangende resultaten op drie domeinen: het bepaalt de systeemkenmerken van natuurlijke taal als een soort algebra, het levert een logica en daarmee ook een ontleed- en/of combineerstrategie, en het levert een systematische semantische representatie. Dit totaalpakket maakt de categoriale invalshoek tot een bijzonder aantrekkelijke positie in de automatische ontleding van natuurlijke taal. Voor een ontleder van het Nederlands is extra aantrekkelijk dat de herleving van categoriale grammatica als instrument van taalkundige analyse onmiskenbaar polderlandse wortels heeft, in de dissertaties van Hoeksema (1984), Zwarts (1986) en Moortgat (1988), en haast het complete oeuvre van Van Benthem (bijvoorbeeld 1991).

*Combinatorische categoriale grammatica*

Eveneens van Nederlandse makelij zijn enkele taalkundige verschijnselen die de hierboven beschreven ruggengraat van de categoriale systematiek lijken te overbelasten. Uiteraard gaat het hier weer vooral om de verbale verstrengeling: de zogenaamde rode volgorde in de werkwoordelijke eindgroep en de daaruit voortvloeiende kruisende verbindingen tussen werkwoorden en objecten. In paragraaf 3 wordt nader ingegaan op de verantwoording van de welhaast unieke werkwoordsgroepen in Delilah. Van de werkwoordelijke eindgroep is overtuigend aangetoond dat ie niet contextvrij is, en dus bijvoorbeeld niet volledig te vangen is in eenvoudige herschrijfregels (Huybregts 1984). De Lambek-rekening kan slecht uit de voeten met dergelijke patronen. Wie de verbale verstrengeling met inachtneming van de Lambek-rekening te lijf wil gaan, moet oftewel het verschijnsel behandelen als een lexicaal gezwel, ofwel de voorzieningen van de Lambek-grammatica dermate uitbreiden dat de rekeningen bijna almachtig worden (Moortgat 1988, Moortgat 1997).

In de opbouw van de categoriale grammatica van Delilah is er voor gekozen de kenmerkende structuren van de Nederlandse werkwoordelijke verstrengeling te beschouwen als één van de primaire, 'gewone' opties voor verketening. Het grammaticamodel is zo opgezet dat de rode volgorde en de kruisende afhankelijkheden in het hart van de grammatica worden vastgelegd. Om te zorgen dat een rekening die voorziet in werkwoordelijke verstrengeling niet triviaal wordt, moet een deel van de Lambekiaanse logica worden opgegeven. Met name regel (16) is dan niet te handhaven. Daarmee vervalt de zogenaamde structurele volledigheid die impliceerde dat elk afleidbare zin talloze deducties had. En daarmee verdwijnt ook het glijdende karakter van de notie *constituent*: die wordt in een grammatica die op niet-triviale wijze Nederlandse verstrengeling door de voordeur binnenhaalt, in ere hersteld.

Al vanaf Ades en Steedman (1980) is er een bescheiden subcultuur geweest van zogeheten *combinatorische categoriale grammatica*, die in plaats van algemene combinatorische regels een veel gerichter stelsel van operaties hanteerde, mogelijk zelfs constructie- en/of taalspecifiek. Steedman (2000) levert een recente beschrijving van deze aanpak. Dit leidt tot een andere algebra, en dus ook tot een andere logica. Het leidt niet tot een andere interpretatie van een afleiding en dus evenmin tot een andere semantiek. Ook de combinatorische variant van categoriale grammatica levert betekenissen in functie van de afleiding.

Delilah profiteert derhalve aan de ene kant van de 'categoriale onderneming' maar kiest aan de andere kant voor een eigenzinnige, het Nederlands toegewijde combinatoriek. Delilah geeft flexibiliteit op, maar wint aan rigiditeit en dus aan beheersbaarheid. Een rechtstreeks gevolg van deze aanpak is dat in Delilah de verhouding tussen afleiding en interpretatie minder rechtstreeks is dan in een standaard categoriale grammatica. Dat is in paragraaf 1 al aan de orde gesteld.

De categoriale grammatica van Delilah is beschreven in Cremers (1993:hfdst.1) en daarna *categoriale lijst grammatica* of *lineaire categoriale grammatica* gedoopt. Net als Moortgat (1997) beschrijft voor multimodale categoriale grammatica, maakt categoriale lijst grammatica gebruik van een serie verschillende verketeningsoperatoren. Elke verketeningsoperator legt in een apart en precies stramien vast welke categorieën op welke wijze kunnen worden gecombineerd. Welke vorm van verketening moet worden gebruikt, hangt af van een lexicale index op het argument 'onder de deelstreep' dat de inzet is van

de verketening: het argument dat wordt weggestreept. Zo'n gespecialiseerde verketeningsoperator is als volgt te omschrijven:

(19) $a/b_i \bullet_i b/c \quad \Rightarrow \quad a/c$
als *a/b*$_i$ en *b/c* aan bepaalde eisen voldoen.

Taalkundig gezien gaat het er natuurlijk om dat het aantal verschillende operatoren beperkt is en dat de voorwaarden voor verketening 'redelijk' en formaliseerbaar zijn. Delilah gebruikt ongeveer tien verschillende operatoren om het Nederlands af te dekken. Enkele van deze operaties worden in paragraaf 3 omschreven en gemotiveerd.

Tot de algemene beperkingen op verketening in Delilah hoort dat de *b* in (19) niet zelf een gerichte deling herbergt, maar altijd een enkelvoudige categorie is. Om deze reden zijn categorieën in de Delilah-grammatica vrij plat. Ze bestaan uit drie delen: een hoofd, een lijst van linker argumenten, en een lijst van rechter argumenten. Een typische categorie voor een verbogen vorm van *dwingen* zou kunnen zijn:

(20) $s\backslash[np_i]/[np_i, vpt_j]$, de categorie die *s* introduceert en links gedeeld wordt door *np*$_i$ en rechts door *np*$_i$ en *vpt*$_j$

Deze categorie drukt uit dat bijvoorbeeld *dwingt* het (syntactische) hoofd is van een zin en drie argumenten neemt: een naamwoordsgroep (als subject), nog een naamwoordsgroep (als object) en een infinitief-complement met *te*. Elk van deze argumenten is geïndexeerd voor de verketeningsoperatie waaronder ze 'weggestreept' kunnen worden. De indexen betekenen niks maar leggen alleen een pakket(je) van eisen vast waaronder verketening met een categorie van de gevraagde soort kan plaatsvinden. De lijst werkt als een stapel: het *vpt*-argument kan pas afgehandeld worden als de rechtse *np* al is verdwenen. Zo is gecodeerd dat in de reeksvorming (de zinsbouw) het nominale object van *dwingt* rechts dichterbij staat dan het infinitief-complement. De volgorde in die lijsten ligt lexicaal vast, en deze volgorde van de categoriesymbolen komt overeen met de lineaire ordening van de corresponderende woordgroepen in de reeks. De lijsten kunnen bij verketening wel met andere samengaan maar worden intern nooit overhoop gehaald. Hieronder staan drie verketeningen die tezamen een zin van het type *NP dwingt NP te VP* (bijvoorbeeld: *de zin dwingt het gedicht te drentelen*) kunnen afleiden; [] staat voor de lege lijst.

(21)

| | | | | |
|---|---|---|---|---|
| $s\backslash[np_0]/[np_0, vpt_6]$ | $\bullet_0$ | $np\backslash[]\ /[]$ | $\Rightarrow$ | $s\backslash[np_0]/[vpt_6]$ |
| *dwingt* | | *het gedicht* | | *dwingt het gedicht* |
| $np\backslash[]/[]$ | $\bullet_0$ | $s\backslash[np_0]/[vpt_6]$ | $\Rightarrow$ | $s\backslash[vpt_6]$ |
| *de zin* | | *dwingt het gedicht* | | *de zin dwingt het gedicht* |
| $s\backslash[vpt_6]$ | $\bullet_6$ | $vpt_6\backslash[]/[]$ | $\Rightarrow$ | $s\backslash[]/[]$ |
| *de zin dwingt het gedicht* | | *te drentelen* | | *de zin ... te drentelen* |

Dergelijke regels sturen het unificatieproces dat in de eerste paragraaf is beschreven, als volgt. Elke categorie in een verketening is deel van, en verwijst naar, een graaf van lexicale, syntactische en semantische kenmerken, als aangeduid in paragraaf 1. Een graaf G1 unificeert met een subgraaf van G2 precies en alleen dan als de categorie van G2 – de ont-

vangende graaf – kan worden verketend met de categorie van G1. Lukt dat, kan de afleiding verder. Mislukt die unificatie, blokkeert het proces. De categoriale grammatica bepaalt dus welke templaten in welke fase van een afleiding aan unificatie worden onderworpen. De categorie voert zo boekhouding en agenda over de combinatorische opties van de resultaatgraaf.

De procedures volgens welke Delilah structuur en interpretatie van een Nederlandse zin berekent, zijn schatplichtig aan meerdere, deels convergerende kaders in de moderne taalkunde. *Categoriale unificatiegrammatica* (bijvoorbeeld Zeevat 1988 en Bouma 1993) was de eerste verbeelding van de gedachte dat unificatie van complexe symbolen door categoriale procedures kon worden gestuurd. De enting op de (hierboven al genoemde) combinatorische categoriale grammatica brengt het systeem in verband met de eigenschappen van *boomaanhechtingsgrammatica's* (TAG). Volgens Joshi en anderen (1991) zetten deze grammatica's de standaard voor een klasse van *mild-contextgevoelige* systemen: grammatica's die krachtiger zijn dan contextvrije formalismes maar niet het volledige bereik van contextgevoeligheid uitbuiten. Cremers (1999) betoogt dat ook lineaire categoriale grammatica tot deze klasse gerekend dient te worden.

## 3 Verschijnselen en analyses

*Verstrengeling*

Ontleders van het Nederlands zijn haast verplicht zich te meten aan werkwoordelijke verstrengeling. Hoeksema (1981) en Hoekstra (1981) leverden de eerste categoriale visies op dit wereldberoemde verschijnsel. Nadien heeft eenieder die de Poolse logica toegedaan was, met de verstrengeling geworsteld.

De categoriale grammatica waarop Delilah draait, voorziet in het soort operaties dat nodig is om de variatie in werkwoordelijke eindgroep en de ermee samenhangende 'kruisende afhankelijkheden' te lijf te gaan. Het belangrijkst is hierbij dat argumenten en complementen precies in verband worden gebracht met het werkwoord dat hun thematische bijdrage regelt op een wijze die recht doet aan de syntactische geleding. In Delilah wordt de werkwoordsgroep gevormd alvorens aan de valentie van deze groep tegemoet wordt gekomen. Bijgevolg bouwt de ontleder gaandeweg de volgende structuur:

(22) ...[ik [ Jeroen [ elke geleerde [[[heb helpen] leren] programmeren]]]]

Dit bouwsel omvat onder meer de constituent *[elke geleerde [[[heb helpen] leren] programmeren]]*. De graaf van deze deelstructuur omvat een LOG-veld waarin de rol van de kwantor *elke geleerde* als patiens van *leren* en agens van *programmeren* is vastgelegd (zie paragraaf 1). Dit verband illustreert de notie 'kruisend': het loopt dwars 'door' het verband tussen bijvoorbeeld *Jeroen* en *helpen*. Per saldo is de representatie van de hele zin zo dat ondermeer de volgende zinnen er berekenbare semantische gevolgen van zijn.

(23) elke geleerde kan programmeren

(24) ik heb Jeroen geholpen
(25) Jeroen heeft elke geleerde iets geleerd

Dergelijke resultaten worden geboekt door de verschijnselen in de Nederlandse werkwoordsgroep af te dekken met behulp van een beperkt aantal combinatorische regels. Deze regels leggen precies die verhouding tussen linkse en rechtse complementen vast die kruisende verbanden mogelijk maakt onder behoud van predikaat-argument-structuren. Om de semantiek op orde te houden zijn contextgevoelige regels nodig. Eén zo'n regel – die welke de deelstructuur *heb helpen* maakt – zal ik hier toelichten.

De deelstructuur moet in ieder geval zo uit de strijd komen dat de *infinitivum-pro-participio* z'n beslag krijgt en de respectievelijke argumenten en complementen van de twee werkwoorden in de gewenste ordening komen. Het is voorts noodzakelijk dat de deelstructuur wordt gevormd vóórdat de twee samenstellende frasen enig ander argument hebben kunnen wegstrepen: zowel de finiete vorm als de infinitief moeten nog hun oorspronkelijke lexicale categorie hebben en niet bij een andere operatie betrokken zijn geweest. De werkwoordelijke verstrengeling is gevoelig voor het onderscheid tussen lexicale en niet-lexicale categorieën, zoals Houtman (1984) al heeft betoogd. Immers, de VP's *rijmen* en *gedichten maken* hebben in alle opzichten dezelfde categorie, maar zijn niet inwisselbaar in de context van verstrengeling: *heeft willen rijmen* versus **heeft willen gedichten maken*.

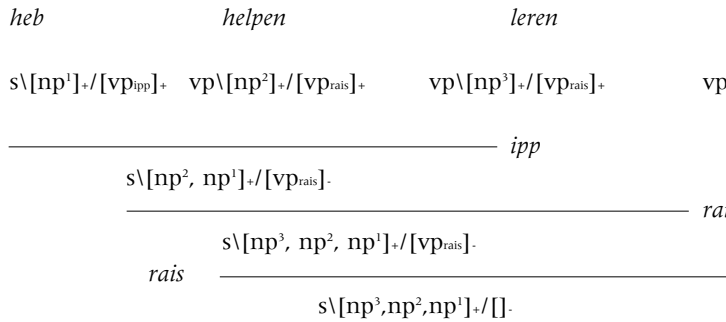
In de Delilah-grammatica wordt de specifieke combinatoriek van hulpwerkwoorden als *heb* ondergebracht in een aparte verketeningsindex (zie paragraaf 2) of samenstellingsmodus. Noem deze modus *ipp*. In de definitie van deze modus is ook vastgelegd dat de betrokken lijsten van argumenten bij het aangaan van de verketening ongeschonden moeten zijn. Hiertoe wordt een tweewaardige index op argumentlijsten bijgehouden: + of -. Daarmee kan zelfs onderscheid gemaakt worden tussen betrokkenheid bij linksgeoriënteerde en rechtsgeoriënteerde verketening. Alle lexicale categorieën starten met index + op beide lijsten. De samenstellingsmodus *ipp* is nu als volgt gedefinieerd, waarbij *[x|Y]* staat voor een lijst met kop *x* en staart *Y* en $X \otimes_{\backslash} Y$ voor de koppeling van lijsten *X* en *Y*:

(26) $a\ \backslash La_{+}\ /\ [vp_{ipp}|Ra]_{X} \quad \bullet_{ipp}\ \ vp\ \backslash Lb_{+}\ /[vp_{j}\ |Rb]_{+} \quad \Rightarrow \quad a\ \backslash Lb \otimes_{+} La\ /[vp_{j}|Rb] \otimes_{-} Ra$

Volgens deze definitie kan een symbool *vp* vooraan in de rechter argumentlijst van een categorie worden geschrapt onder de verketeningsmodus *ipp* mits drie van de vier lijsten index + hebben en de rechterlijst van de tweede categorie als eerste categoriesymbool *vp* heeft. In de resulterende categorie wordt $La_{+}$ achter $Lb_{+}$ geplakt tot de nieuwe onaangeroerde lijst $Lb \otimes_{+} La$. *Ra* komt achter het restant van de andere rechterlijst. De nieuwe lijst van linkse argumenten krijgt index +, want deze lijst is uit de wind gebleven en neemt de maagdelijkheid van de samenstellende lijsten over. De nieuwe rechterlijst daarentegen wordt samengesteld uit lijsten die al aan de tand zijn gevoeld, en krijgt derhalve index -.

De afleiding van (22) neemt nu de volgende vorm aan; *np*'s zijn van nummers voorzien om ze uit elkaar te houden, maar hun toepassingsmodi zijn weggelaten. De overige toepassingsmodi zijn *rais* gedoopt, naar *v-raising*. Die verschilt alleen hierin van *ipp* dat geen argumenttype *vp* in de 'rechtse' categorie is vereist. In plaats van de verketeningsoperator is hier een streep gebruikt. De modus van de verketening is gelijk aan de index op het te schrappen argument.

(27)




De resulterende categorie bindt achtereenvolgens *elke geleerde, Jeroen* en *ik*. Het achterliggende templaat bindt bij unificatie deze naamwoordsgroepen aan hun respectievelijke argumentsposities en thematische rollen.

Op dergelijke wijze is voor elk van de verstrengelingsvormen van het Nederlands – zoals extrapositie, *v-raising* en de derde constructie – een bepaalde modus verantwoordelijk. De modus maakt in alle gevallen deel uit van de categoriale specificatie van het 'hoofdwerkwoord', dat is het werkwoord dat een verbaal complement van enig soort selecteert. Bijgevolg is de afhandeling van de werkwoordelijke eindgroep, evenals vrijwel alle andere aspecten van de grammatica, lexicaal gestuurd.

Hierbij moet worden opgemerkt dat de plaatsing van bepalingen, zowel bijvoeglijke als bijwoordelijke, wordt geregeld door modi die ook op werkwoordelijke verstrengeling van toepassing zijn. Aldus zijn de meeste modi beduidend algemener dan een enkele toepassing doet vermoeden. De aanduidingen *ipp* en *rais* hierboven zijn dan ook eerder misleidend dan karakteristiek.

*Nevenschikking*

De oorsprong van Delilah is de implementatie van een algoritme dat nevenschikking op een buitengrammaticale manier afhandelt. De grondgedachte hier is dat nevenschikking niet door de 'zins'-grammatica wordt geconfigureerd, maar parasiteert op grammatische structuren en daarbij analytische middelen inzet die niet tot het domein van de grammatica behoren. In Cremers (1993) is geprobeerd zo'n benadering van nevenschikking te rechtvaardigen en het algoritme te beschrijven dat deze klus kan klaren voor een categoriale grammatica.

Delilah gaat woeste vormen van non-constituent-nevenschikking te lijf, als in

(28) de aankondiging dat *elke man Agnes het boek met* en *elke vrouw mij enkele gedichten over* de auto wilde geven werd ontkend

De strategie is om het bereik van een nevenschikking – het bepalen van welke frasen binnen en welke buiten de nevenschikking vallen – af te leiden uit overlap van deelontledingen van de zin aan weerszijden van nevenschikkingen. Deze deelontledingen komen tot stand onder de veronderstelling dat de reeks een welgevormde zin betreft. De precieze werking van dit algoritme in relatie tot de complexiteit van de gehele ontleder is besproken in Cremers en Hijzelendoorn (1997). Cremers (1993:hfdst. 2) verantwoordt de

herleiding van elke nevenschikking op zinsnevenschikking.

Voor elke semantische operatie rond nevenschikking is het van wezenlijk belang dat de aard en het bereik van de nevenschikking wordt bepaald zonder extra druk op de grammatische analyse. De buitengrammatische benadering van Delilah zorgt hiervoor. De herleiding van alle nevenschikking op zinsnevenschikking maakt het in ieder geval mogelijk de semantische bijzonderheden van elke nevenschikking te bepalen en de interpretatie daarbij aan te passen.

*Collocaties*

Elke taal is vergeven van woordgroepen met gespecialiseerde betekenissen. Lexicaal gedreven grammatica's als die van Delilah lenen zich in beginsel goed voor het compositioneel vastleggen en volgen van de lotgevallen van dergelijke collocaties in zinsverband. De categoriale grondslag van de grammatica brengt mee dat hoofden van woordgroepen hun volledige projectie meevoeren in hun templaat. Een woord van de categorie *a/b* introduceert een templaat waarin al relevante eigenschappen van het *b*-argument zijn opgevoerd, zoals het type, de bijdrage van z'n betekenis aan de betekenis van het geheel en bijvoorbeeld casus. Bijgevolg is het niet bijzonder om dit argument nog verder uit te werken en de semantische waarde van dit geheel te fixeren. Bijvoorbeeld: het templaat van het transitieve *hebben* wordt in het lexicon gekoppeld aan het templaat van *honger*, en de betekenis van de combinatie wordt gespecificeerd in termen van *be hungry*. Combinatorisch gedraagt het templaat zich verder volstrekt normaal.

Alle collocaties die een lexicale kern hebben, kunnen op een vergelijkbare wijze hun beslag krijgen. Het is evenwel van belang na te gaan of deze handelswijze gevolgen heeft voor de meerduidigheid van de analyse. Wordt het aantal lexicale vertegenwoordigers van een woord door de specificatie van collocaties niet zo groot dat de economie van de ontleding wordt aangevreten? Dat valt mee. De ontleding beziet in de combinatorische fase alle verschillende categorieën van een woord of woordgroep. Collocaties verschillen per definitie niet in categorie van 'open' verbindingen. Pas als een categorie als combinatorisch relevant is geselecteerd, komen de onderscheiden templaten die deze categorie voeren aan bod. Het is hierbij wel haast onvermijdelijk dat een verbinding zowel gecolloceerd als open wordt geïnterpreteerd: de zin *ik heb honger* krijgt zowel de lezing *i'm hungry* als *i possess hungryness*. De laatste lezing kan wel in algemene termen worden uitgesloten door bijvoorbeeld abstracta te weren als object van *hebben*/bezitten, maar dat zou getuigen van metafysisch opportunisme.

*Eilanden*

Zoals hierboven al is gesteld, is de eerste semantische representatie in Delilah – na voltooiing van de categoriale combinatoriek en de unificatie van templaten – een ondergespecificeerde opslag van λ-termen in het LOG-veld. Deze opslag weerspiegelt de combinatorische structuur: de argumenten van een functor worden bij unificatie opgeslagen 'bij' die functor. Er is een apart, postderivationeel algoritme dat vervolgens deze opslag uitschrijft door een combinatie van twee operaties: de conversie die typerend is voor λ-termen, en de verplaatsing 'omhoog' van opgeslagen termen. Hierboven is al aangegeven dat deze twee-

trapsraket goed vergelijkbaar is met de *Quasi Logical Form* van Alshawi en anderen (1992) en met verdere onderspecificatiebenaderingen, zoals Muskens (2001) en Kempson, Meyer-Viol en Gabbay (2001).

Het postderivationele algoritme draagt zorg voor semantische filtering. Het algoritme moet bijvoorbeeld zo werken dat een semantisch element niet aan een voor dat element relevant eiland kan ontsnappen. De concrete structuren die Delilah hanteert, zijn te groot om hier zinvol te bekijken; vandaar dat hier een sterk vereenvoudigd voorbeeld wordt gehanteerd. Stel dat (29) na categoriale combinatoriek en bijbehorende unificatie een templaat oplevert waarin (30) de waarde in het hoogste LOG-veld vormt.

(29) Iedereen:NP1 ontkende:V1 dat er een gedicht:NP2 sliep:V2
(30) [NP1 [ [ NP2] : V2 ] ] : V1

Deze structuur moet dan zo gelezen worden dat V1 de semantiek van het hoofdwerkwoord vertegenwoordigt. Alle andere semantische bijdragen maken deel uit van de opslag van V1. Die opslag bevat daarom twee grootheden: de semantische bijdrage van het subject en de semantische bijdrage van het object. De laatste bestaat zelf weer uit de semantiek van het ingebedde gezegde met in z'n opslag de bijdrage van het ingebedde subject.

Het uitschrijfalgoritme werkt de diepte in, en begint met het toepassen van de laagste opslag op de bijbehorende grootheid, in dit geval de toepassing van de kwantor NP2 op V2. Het algoritme beziet tevens of de grootheid NP2 naar de eerst hogere opslag verplaatst kan worden, om een alternatieve bereikstoewijzing af te dwingen. Het algoritme wroet hiervoor in de overige informatie van het templaat van (29): maakt de constituent die NP2 representeert, deel uit van de structuur die als eiland is gespecificeerd en is NP2 zodanig dat het voor deze eilandeigenschap gevoelig is? In dit geval zal de constituent waarvan *ontkende* het hoofd is, door een lexicale specificatie op dit werkwoord gemarkeerd zijn als een negatief eiland. De existentiële kwantor verliest daardoor z'n dynamiek, en moet in de kelder blijven. Groot bereik is uitgesloten. De enige lezing ontstaat door successievelijke toepassing van de opgeslagen λ-termen als weergegeven in (31), met groot bereik voor *iedereen*.

(31) NP1( V1( NP2( V2 ) ) )

Indien het hoofdwerkwoord niet voor negativiteit zou zijn gemarkeerd, was een grootbereik-lezing voor het ingebedde subject tot stand gekomen via een transport van NP2 naar de hogere opslag, als in (32). De corresponderende lezing zou zijn verkregen met de successievelijke toepassing :

(32) [ NP2 NP1 [ [ ] : V2 ] ] : V1
(33) NP2( NP1( V1 ( V2 ) ) )

De moraal is dat toegang tot alle relevante gegevens verrekening van eilanden mogelijk maakt. Omdat niet gegarandeerd kan worden dat alle gegevens ter beschikking zijn voor het unificatieproces is afgerond, is dit een operationeel argument voor 'late' interpretatie.

Blijft de vraag of alle kenmerken van eilanden via unificatie beslisbaar zijn op basis van lexicale specificaties. Deze vraag is natuurlijk niet veel anders dan de veel algemenere vraag

of relevante semantische eigenschappen van constituenten berekenbaar zijn uit samenstelling en inbedding. Dit is wat het beginsel van compositionaliteit zou moeten afdwingen. Als een inrichting van de grammatica deze eigenschap niet heeft, is dat model niet erg geschikt voor computationele toepassingen. Overigens kunnen eilandachtige beperkingen op combinatoriek ook door extra operatoren in de categoriale grammatica worden opgelegd; zie voor zo'n aanpak bijvoorbeeld Hepple (1990) en Bernardi (2002). Het bijzondere van de aanpak in Delilah is dat vanwege het opslagsysteem voor λ-termen syntactische eiland-effecten anders behandeld (kunnen en moeten) worden dan semantische.

## 4 Problemen en oplossingen

Niets drukt een taalkundige meer op de gaten in z'n taalbeschouwing dan het bouwen van een model dat het moet doen. Bijgevolg is het na jaren kleien, vijlen en lijmen ook geen probleem om huidige tekortkomingen van het onderhavige model op te sommen. Naast informatiekundige gebreken, zijn er talrijke voorbeelden van gebrekkige grammatica, noodvoorzieningen, blinde vlekken en onbetreden velden. Ik zal hier enkele van de voornaamste dekkingsproblemen noemen.

Alhoewel het coördinatiealgoritme de wieg is van het ontleedsysteem, is er nog geen procedure ontwikkeld om meervoudige nevenschikking te lijf te gaan. De grond voor dit gebrek is helder. Het coördinatiealgoritme is extragrammaticaal omdat de notie 'coördinaat' of 'gecoördineerd zinsdeel' geen categoriale status heeft. Wat gecoördineerd is, laat zich niet aflezen aan de interne bouw van het zinsdeel, maar uitsluitend aan het geheel van de omgeving (zie Cremers 1993). Het coördinatiealgoritme van Delilah buit deze spanning tussen inbedding en interne structuur van nevenschikking volledig uit. Het neemt daarbij evenwel de zinsgrenzen en de relatieve positie van het nevenschikkend element als ankers, omdat de categoriale structuur geen aanknopingspunten biedt. Dit gaat onvermijdelijk ten koste van de nauwkeurigheid en doelgerichtheid in de analyse, zoals is uiteengezet in Cremers en Hijzelendoorn (1997). Dit gebrek aan gerichtheid zou exploderen indien een coördinatie syntactisch en semantisch moet worden opgelost met behulp van nevengeschikte structuren in de omgeving: het anker vindt dan geen grond. Bijgevolg zou in een configuratie met twee nevenschikkingen bijna elke beperking op de klasse van mogelijke analyses weg vallen. Er is geen reden om aan te nemen dat het algoritme niet tot een juiste analyse kan komen, maar efficiëntie zal dan ver te zoeken zijn.

Het coördinatiealgoritme is voorts nog niet bestand tegen ellipsis in nevenschikkend verband. Het gaat hier om elliptische constructies als *gapping* (34) en vergelijking (35):

(34) Jan verwees mij naar de kommissie en jou naar het bestuur
(35) Hij heeft vaker gelogen dan enig ander vóór hem

Dat comparatiefconstructies als nevenschikkingen beschouwd moeten worden, is overtuigend betoogd in Hendriks (1995). Dat het coördinatiealgoritme niet met ellipsis overweg kan, is evenwel geen conceptueel maar een procedureel probleem. Aangezien het coördinatiealgoritme ontworpen is voor non-constituent-coördinatie, is het ontbreken van een module voor elliptische reconstructie allerminst principieel.

Een belangrijke flauwte in de semantische component van Delilah zit in de structuur van predicatie. Nu wordt daarvoor het stramien van de *n*-plaatsige relatie gebruikt. Een dergelijke semantiek past bij een beperkte ontologie. Voor processen als nominalisatie maar ook voor de juiste analyse van bijvoorbeeld collectiviteit en distributiviteit is een directe verwijzing naar gebeurtenissen en toestanden op objectniveau gewenst, zo niet noodzakelijk. Dat roept om een semantiek waarbij zin (36) eerder als (37) dan als (38) zal worden beschreven.

(36) Elke lobbyist heeft een politicus gebeld
(37) $\forall x\, L(x) \Rightarrow \exists y\, P(y) \,\&\, Perf.B(x,y)$
(38) $\forall x\, L(x) \Rightarrow \exists e\, B(e) \,\&\, \exists y\, P(y) \,\&\, Agens(e,x) \,\&\, Patiens(e,y) \,\&\, Perf(e)$

De taalkundige perspectieven van zo'n ('davidsoniaanse') semantiek zijn beschreven in Parsons (1990). Belangwekkende toepassingen zijn te vinden bij bijvoorbeeld Schein (1995) en Doetjes en Honcoop (1998). De omzetting naar een *event*-gestuurde representatie heeft ook belangrijke voordelen voor de generator: de formele adresseerbaarheid van gebeurtenissen en standen-van-zaken maken het systeem beter geschikt om databanken te verwoorden.

Een geniepige tekortkoming in de huidige grammatica van Delilah is dat rechtse verplaatsing, extrapositie of rechtswaartse dislocatie nog niet stabiel is afgehandeld. Dit heeft een heldere systematische oorzaak. Rechtswaartse dislocatie heeft doorgaans betrekking op bijvoeglijke nabepalingen. Deze kunnen op het einde van de clause van hun doelwit voorkomen.

(39) Ik heb de *man* proberen op te bellen *die de prins beledigd heeft*
(40) De stakker die het *boek* had gekocht *over de jonge Hegel* is gisteren opgenomen.

In de voorbeeldzinnen zijn bepaling en doelwit gecursiveerd. Tussen beide kan materiaal staan dat op z'n best in afgeleide zin betrokken is bij de semantische verhouding tussen bepaling en nomen. Als de bijvoeglijke bepaling opgevat moet worden als een functor op zoek naar een argument – dat is de standaard typenlogische benadering maar zie hieronder – is het traject naar het argument voor de adjunctieve functor semantisch irrelevant. Van samenstelling als combinatorisch proces kan hier dus geen sprake zijn. Samenstelling impliceert immers een semantische ordening die hier volstrekt afwezig is: tussen *op te bellen* en *die de prins beledigd heeft* bestaat op z'n best een afgeleide relatie. In deze observatie zit tevens de kern van de oplossing verscholen. Tussen de naamwoordelijke groep en z'n rechteromgeving bestaat wel een semantische relatie die categoriale samenstelling kan velen. Als niet het adjunct maar de naamwoordsgroep de 'zoekende' functor is, kan rechtse samenstelling de positie voor het adjunct via lijstvermenging in stelling brengen. Deze suggestie is al te vinden bij Janssen (1983). Hier is wat er combinatorisch ongeveer zou moeten gebeuren om *elke dichter straffen die rijmt* als een samenhangende werkwoordsgroep te analyseren:

(41)

| | | | | |
|---|---|---|---|---|
| np/n • | n/rel • | vp\np • rel | ⇒ np/rel • vp\np • rel ⇒ | vp/rel • rel ⇒ vp |
| *elke* | *dichter* | *straffen die rijmt* | | |

De Delilah-grammatica zou met deze overgangen allerminst problemen hebben; sterker: die leent zich buitengewoon goed voor de 'disharmonische' samenstelling die hier gewenst lijkt. Maar de behandeling van (bijvoeglijke na-)bepalingen als argumenten – eerder dan als functoren – strookt nog niet met de benaderingswijze van adjuncten die in Delilah is ingebakken.

Het relevante alternatief is al eerder – op andere maar niet wezenlijke afwijkende gronden – geformuleerd door Bouma en Van Noord (1994). Zij brengen de grammatische optie om adjuncten te behandelen als argumenten onder in een parseerstrategie die een explosie van ambiguïteit tegengaat.

Deze strategie zou behalve het probleem van de rechtse dislocaties ook enkele andere tekortkomingen van lexicaal gestuurde grammatica's oplossen. Het is in Delilah bijvoorbeeld niet mogelijk *wh*-extractie van adjuncten correct te interpreteren. De zin

(42) Waar denk jij dat hij mij wou onder brengen?

kan Delilah alleen interpreteren als een vraag naar de locatie van *denken*. Als daartegen *waar* ook opgevat kan worden als een argument van *onderbrengen*, kan het vraagelement op dezelfde wijze bij de ingebedde zin betrokken worden als *wie* in

(43) Wie denk jij dat hij hier wou onder brengen?

Er zijn andere eigenschappen van adjuncten die bij een benadering als argument verloren dreigen te gaan. Zo laat zich de betrekkelijk vrije positie van adjuncten ten opzichte van andere zinsdelen uitstekend beschrijven als een combinatorische optie van het adjunct als functor. Bezie bijvoorbeeld de reeks

(44) Ik probeer Jan vrijwillig het boek voor Agnes te laten kopen
(45) Ik probeer Jan het boek vrijwillig voor Agnes te laten kopen
(46) Ik probeer Jan het boek voor Agnes vrijwillig te laten kopen

Stel dat *vrijwillig* hier als een functor van de categorie $vp\backslash[]_+/[vp_4]_+$ wordt beschouwd. De modus *4* laat allerhande argumentlijsten bij de secundaire categorie toe. Het is dezelfde modus die ook de zogenaamde *derde constructie* mogelijk maakt: gedeeltelijke verstrengeling, bijvoorbeeld: *... mij geprobeerd een wrak aan te smeren*. Onder een dergelijke categorisering van het adjunct als functor over z'n relevante omgeving laten zich de plaatsingsopties correct afleiden. Deze aanpak komt uiteraard in het gedrang indien *vrijwillig* als argument aan bijvoorbeeld *kopen* wordt gekoppeld. In plaats van een 'open' categorisering moeten dan de verschillende ordeningen op de een of andere manier in de lexicale categorieën van dit werkwoord worden opgeslagen. Dat is geen bijdrage aan de doelmatigheid.

Per saldo lijkt de behandeling van adjuncten als argumenten de prijs die lexicaal gestuurde ontleedsystemen betalen voor combinatorische volledigheid. Wellicht dat adjuncten dan wel als bijzondere argumenten moeten worden bestempeld, met een vlottende positie in de argumentlijsten. Een alternatief is om categorieën niet één maar meerdere, combinatorisch gescheiden stapels van argumenten mee te geven: een stapel voor gewone argumenten, een stapel voor *wh*-argumenten en een stapel voor adjuncten. Deli-

lah en andere op betekenis georiënteerde ontleedmachines moeten vroeg of laat voor een van deze stategieën kiezen. Aldus blijft men doende.

## Bibliografie

**Ajdukiewicz, Kazimierz (1935).** Die syntaktische Konnexität. *Studia Philosophica* 1, 1-27.

**Alshawi, Hiyan (red.) (1992).** *The Core Language Engine*. Cambridge, Mass.: MIT Press.

**Benthem, Johan van (1991).** *Language in Action*. Amsterdam: North-Holland.

**Bernardi, Raffaella (2002).** *Reasoning with Polarity in Categorial Type Logic*. Dissertatie, Universiteit Utrecht.

**Bouma, Gosse (1993).** *Nonmonotonicity and Categorial Unification Grammar.* Rijksuniversiteit Groningen, Groningen dissertations in linguistics.

**Bouma, Gosse en Gert-Jan van Noord (1994).** Constraint based categorial grammar. *Proceedings 32nd Annual meeting of the ACL*, 147-154.

**Buszkowski, Wojchiech (1988).** Generative Power of Categorial Grammars. In: R.T. Oehrle e.a. (eds), *Categorial Grammars and Natural Language Structures*. Dordrecht: Reidel, 69-94.

**Carpenter, Bob (1997).** *Type-logical Semantics*. Cambridge, Mass.: MIT Press.

**Cooper, Robin (1975).** *Montague's semantic theory and transformational syntax*. PhD dissertatie, University of Massachusetts.

**Cremers, Crit (1993).** *On parsing coordination categorially*. Universiteit Leiden, HIL dissertation.

**Cremers, Crit (1999).** A Note on Categorial Grammar, Disharmony and Permutation. *Proceedings of EACL '99*. ACL, 273-275.

**Cremers, Crit (2001).** Modal Merge and Minimal Move for Dislocation and Verb Clustering. *Journal of Language and Computation* 1:5.

**Cremers, Crit, en Maarten Hijzelendoorn (1997).** Pruning Search Space for Parsing Free Coordination in Categorial Grammar. *International Workshop on Parsing Technologies. Proceedings 1997*. Cambridge, Mass.: MIT, 42-53.

**Doetjes, Jenny en Martin Honcoop (1998).** The Semantics of Event-related Readings. A Case for Pair-quantification. In: Anna Szabolcsi (red.), *Ways of Scope Taking*. Dordrecht: Kluwer, 263-310.

**Heim, Irene en Angelika Kratzer (1998).** *Semantics in Generative Grammar*. Oxford: Blackwell.

**Hendriks, Herman (1993).** *Studied Flexibility.* Universiteit van Amsterdam, ILLC Dissertation Series.

**Hendriks, Petra (1995).** *Comparatives and Grammar*. Dissertatie, Rijksuniversiteit Groningen.

**Hepple, Mark (1990).** *The Grammar and Processing of Order and Dependency. A Categorial Approach*. Dissertatie, University of Edinburgh.

**Hoeksema, Jack (1981).** Verbale verstrengeling ontstrengeld. *Spektator* 10, 221-249

**Hoekstra, Teun (1981).** The Base and the Lexicon in Lexical Grammar. In: Saskia Daalder en Marinel Gerritsen (red.), *Linguistics in the Netherlands 1981*. Amsterdam: North Holland, 93-101.

**Houtman, Joop (1984).** Een kategoriale beschrijving van het nederlands. *Tabu* 14, 1-27.

**Huybregts, Riny (1984).** The weak inadequacy of context-free phrase structure grammars. In: G.J. de Haan e.a. (red.), *Van Periferie naar Kern*. Dordrecht: Foris, 81-99.

**Janssen, Theo (1983).** *Foundations and Applications of Montague Grammar*. Dissertatie, Universiteit van Amsterdam.

**Joshi, Aravind, K. Viyai-Shanker en David Weir (1991).** The Convergence of Mildly Context-Sensitive Formalisms. In: Peter Sells, Stuart Shieber en Tom Wasow (red.), *Processing of Linguistics Structure*. Cambridge, Mass.: MIT Press, 31-81.

**Keller, Bill (1993).** *Feature Logics, Infinitary Descriptions and Grammar*. Stanford, CSLI.

**Kempson, Ruth, Wilfried Meyer-Viol en Dov Gabbay (2001).** *Dynamic Syntax: The Flow of Language Understanding*. Oxford: Blackwell.

**Lambek, Joachim (1958).** The Mathematics of Sentence Structure. *American Mathematical Monthly* 65, 154-170.

**Montague, Richard (1973).** The Proper Treatment of Quantification in Ordinary English. In: J. Hintikka, J. Moravcsik en P. Suppes (red.), *Approaches to Natural Languages*. Dordrecht: Reidel, 221-242.

**Moortgat, Michael (1988).** *Categorial Investigations*. Dordrecht: Foris.

**Moortgat, Michael (1997).** Categorial Type Logics. In: J. van Benthem en A. ter Meulen (red.), *Handbook of Logic and Language*, Amsterdam: North-Holland, 93-178.

**Muskens, Reinhard (2001a).** λ-Grammars and the Syntax-Semantics Interface. In: R. van Rooy en Martin Stokhof (red.), *Proceedings of the Thirteenth Amsterdam Colloquium*. Universiteit van Amsterdam, ILLC, 150-155.

**Muskens, Reinhard (2001b).** Talking about Trees and truth-Conditions. *Journal of Logic, Language and Information* 10, 417-455.

**Parsons, Terence (1990).** *Events in the Semantics of English*. Cambridge: MIT Press.

**Partee, Barbara (1992).** Syntactic category and semantic type. In: Michael Rosner en Roderick Johnson (red), *Computational linguistics and formal semantics*. Cambridge University Press, 97-126.

**Penn, Gerald (2000).** *The Algebraic Structure of Attributed Type Signature*. Dissertatie, Carnegie Mellon University.

**Schein, Barry (1993).** *Plurals and Events*. Cambridge, Mass.: The MIT Press.

**Steedman, Mark (1990).** Gapping as Constituent Coordination. *Linguistics and Philosophy* 13, 207-263.

**Steedman, Mark (1996)**. *Surface Structure and Interpretation*. Cambridge: The MIT Press.

**Steedman, Mark (2000).** *The syntactic process*. Cambridge, Mass.: The MIT Press.

**Zeevat, Henk (1988).** Combining Categorial Grammar and Unification. In: Uwe Reyle en Christiaan Rohrer (red.), *Natural Language Parsing and Linguistic Theories*. Dordrecht: Reidel, 202-229.

**Zwarts, Frans (1986).** *Categoriale grammatica en algebraïsche semantiek*. Dissertatie, Rijksuniversiteit Groningen.

# Boekbeoordelingen

**S. Gillis & A. Schaerlaekens (red.).** *Kindertaalverwerving. Een handboek voor het Nederlands*. Groningen: Martinus Nijhoff, 2000. XIV + 563 blz. ISBN 90 68 90503 1 EUR 42,00.

Al enige tijd geleden verscheen dit zeer uitgebreide handboek over kindertaalverwerving, dat we kunnen beschouwen als een vervolg op het bekende, in 1987 gepubliceerde boek *De taalverwerving van het kind* van Schaerlaekens en Gillis. Dat deze auteurs tien jaar later in de hoedanigheid van redacteuren hebben gekozen voor een veel uitgebreider boek met zeventien contribuanten, heeft verschillende redenen. Sinds 1987 is er enorm veel gepubliceerd over taalverwerving, wat onmiddellijk is vast te stellen door de indrukwekkende bibliografie van 42 pagina's door te bladeren. Het vakgebied is dan ook nog nauwelijks te overzien door één of twee onderzoekers. Een andere reden is dat door de thematische aanpak elke auteur zijn onderwerp vanuit zijn eigen onderzoeksgebied kan benaderen, zodat veel van de materie uit de eerste hand gepresenteerd kan worden.

Het is niet mogelijk binnen de ruimte die deze bespreking is toegemeten, enigszins uitgebreid op de inhoud van dit omvangrijke boek in te gaan. Ik zal hieronder dan ook kort aangeven waar elk hoofdstuk over gaat en vervolgens wat commentaar geven.

Het boek begint met een zeer algemene Inleiding van Annemarie Schaerlaekens, waarin een kort overzicht wordt gegeven van dit werk en een aantal belangrijke begrippen wordt uitgelegd.

Hoofdstuk 1, geschreven door dezelfde auteur, geeft vervolgens een blauwdruk van de verwerving van het Nederlands en is bedoeld als een eerste, oriënterende kennismaking met het proces van taalverwerving. Aan de orde komen de verschillende aspecten van de moedertaal die in de diverse perioden van de taalverwerving worden geleerd: 1. De prelinguale periode (0-1 jaar), 2. De vroeglinguale periode (1-2,5 jaar; éénwoordfase en twee- en meerwoordfase), 3. De differentiatiefase (2,5-5 jaar; fonologische ontwikkeling, semantiek en woordenschat, syntaxis en morfologie, pragmatische en metalinguïstische aspecten en communicatie-aspecten), en 4. De voltooiingsfase (5-9 jaar).

Het tweede hoofdstuk (auteurs: Steven Gillis en Annick de Houwer) is meer methodologisch van aard. Er wordt uiteengezet op welke manier het feitenmateriaal verzameld wordt, met veel aandacht voor de diverse tests, de manier waarop de jonge proefpersonen worden geselecteerd, de verzamelde gegevens worden vastgelegd, geanalyseerd, enzovoort. Een belangrijk punt is natuurlijk de vraag hoe men achter de intuïties komt van jonge taalgebruikers, die immers nog geen metalinguïstisch bewustzijn hebben ontwikkeld. Een andere moeilijkheid is het bepalen van het tijdstip waarop met een bepaald type observatie moet worden begonnen, aangezien er tussen kinderen behoorlijke verschillen zijn wat betreft de leeftijd waarop zij een bepaalde stap zetten in de taalverwerving. Ook bespreken de auteurs de waarde van twee taalmaten die in de onderzoekspraktijk van kindertaal veel gebruikt worden: de Gemiddelde Uitingslengte (een kwantitatieve maat van (morfo-)syntactische complexiteit) en de Type-Token Ratio (een kwantitatieve maat van lexicale diversiteit).

Vervolgens wordt in de hoofdstukken 3-11 telkens een aspect van taalverwerving uitgediept. In hoofdstuk 3 richten de auteurs, Jeanette van der Stelt en Florien Koopmans-

van Beinum, zich op de klankperceptie en -productie in het eerste levensjaar. Het bestaat uit drie delen. Het eerste deel gaat over het perceptiesysteem van het kind (de geluids- en spraakwaarneming). Het tweede deel bespreekt zijn productiesysteem (de geluids- en spraakproductie). In het derde deel staat het wederzijdse leerproces in de moeder-kindinteractie centraal. Omdat is vastgesteld dat een foetus van vijf maanden al simpele geluidsaspecten kan horen, beginnen de auteurs hun verhaal reeds vanaf het moment dat het kind zich nog in de baarmoeder bevindt.

In hoofdstuk 4 wordt door Steven Gillis de fonologische ontwikkeling geschetst, waarbij de auteur zich beperkt tot de klankproducties van het kind. Het receptieve aspect van de fonologische ontwikkeling blijft achterwege. Eerst wordt ingegaan op de segmentele verwerving: het leren van de consonanten en vocalen. Daarbij wordt tevens uiteengezet wanneer een segment daadwerkelijk verworven is. Vervolgens is er aandacht voor de verklaring van de verwervingsvolgorde. Daarna komen de syllaben en woorden aan de orde, alsmede de verschillende fonologische processen (zoals assimilatie en substitutie) die eigen zijn aan de kindertaal. Ten slotte wordt uiteengezet hoe de systematische vervormingen in de kindertaal verklaard kunnen worden.

In het vijfde hoofstuk beschrijven Loekie Elbers en Anita van Loon-Vervoorn de verschillende perioden waarin de ontwikkeling van woordbetekenis en lexicon zich voltrekt. Daarbij is gekozen voor een indeling in drie perioden: de referentiële periode (waarin het kind relaties leert leggen tussen woorden en gebeurtenissen), de denotationele periode (waarin het kind de grenzen leert kennen van woordbetekenissen: welke zaken vormen samen een benoembare klasse en welke niet?), en de 'sense'-periode (waarin het kind leert welke woorden conceptueel bij elkaar horen en op welke manier). Verder wordt in dit deel van het boek de lexicale creativiteit van het kind besproken, alsmede de verwerving van de idiomen van de moedertaal.

Hoofdstuk 6, geschreven door Jacqueline van Kampen en Frank Wijnen, gaat over de grammaticale ontwikkeling van de kindertaal. Daarbij wordt ingegaan op de achtereenvolgende stadia (telegramstijlfase en differentiatiefase) en hun grammaticale kenmerken. Daaronder vallen de morfologie en de syntaxis. Ook de snelheid waarmee de veranderingen gepaard gaan, wordt in ogenschouw genomen. Vervolgens wordt de vraag gesteld hoe zowel de leerstappen als het tempo kunnen volgen uit het taalverwervend vermogen en het taalaanbod. Interessant is de visie van de auteurs op de grammaticale capaciteiten van het kind: in het stadium van de telegramstijl bezit het kind reeds de grammaticale relaties, maar de taalspecifieke markeringen daarvan nog niet. Volgens de auteurs bezit het kind een ondergespecificeerde grammatica. Het gebruikt echter klemtoonpatronen om betekenisonderscheid aan te brengen. In de differentiatiefase is het leren van die taalspecifieke markeringen een kwestie van invullen op lege plaatsen.

Het zevende hoofdstuk is van de hand van Anne Baker, Claudia Blankenstijn en Marja Roelofs en gaat over de pragmatische ontwikkeling, die doorgaat tot minstens het tiende jaar. Eigenlijk gaat deze het hele leven door. Na enkele algemene beschouwingen volgen een beschrijving van de ontwikkeling van communicatieve intenties, een uiteenzetting van de manier waarop een gesprek en een verhaal worden opgebouwd, een bespreking van de wijze waarop kinderen leren om informatie goed over te brengen, en van situaties van taalgebruik zoals liegen en het vertellen van een anekdote, mop, enzovoort. In dit hoofdstuk gaat het vooral om de ontwikkeling van de mondelinge pragmatische vaardigheden. Waar dat mogelijk is, hebben de auteurs de ontwikkeling gekoppeld aan leeftijd.

Hoofdstuk 8, geschreven door Catherine Snow gaat over de rol van taalaanbod en sociale interactie in de taalverwerving. Gekeken wordt

naar het effect van de volgende aspecten: 1. een specifiek, aangepast register door de ouders in hun interactie met het kind, 2. negatieve feedback, 3. kwalitatieve en kwantitatieve verschillen in het taalaanbod, en 4. universele onderdelen van Child Directed Speech (CDS), en 5. taal-, cultuur- en kindspecifieke eigenschappen. Het gebruik van CDS is alleen nuttig als het kind in staat is deze te begrijpen en met zijn eigen uitingen te vergelijken. Voor oudere kinderen kan de áfwezigheid van CDS juist bevorderlijk zijn voor de taalverwerving, doordat er meer van het kind geëist wordt. Kinderen die consequent negatieve feedback krijgen in de vorm van herhalingen van de correcte vorm, leren sneller dan kinderen bij wie dat achterwege blijft. Verder blijkt een kind vlotter de taal te leren naarmate het taalaanbod groter is. Ook de timing van woordgebruik is cruciaal, bijvoorbeeld het noemen van een werkwoord net vóór de actie waarop het werkwoord betrekking heeft. Wat de precieze consequenties zijn van culturele verschillen in de opvoeding voor de taalverwerving, is nog nauwelijks onderzocht.

Hoofdstuk 9 van René Appel en Anne Vermeer gaat over tweedetaalverwerving van kinderen en simultane taalverwerving (het leren van twee moedertalen). De auteurs geven een schets van de fonologische, morfologische, syntactische en lexicale ontwikkeling bij tweedetaalverwerving. Ook is er aandacht voor de pragmatische ontwikkeling. Daarnaast bespreken zij enkele theorieën over T2-verwerving en de factoren die van invloed zijn op dit taalleerproces. Vervolgens gaan zij in op de verwerving van twee moedertalen. Belangrijke verschillen tussen T2- en T1-verwerving zijn onder meer het tempo waarin de desbetreffende taal wordt geleerd en het eindniveau waarop die taal wordt beheerst. Daarnaast zijn er overeenkomsten in de stappen waarin T1- en T2-verwerving verlopen. Bij de theorieën over tweedetaalverwerving gaat het over de visie die een grote rol toekent aan de invloed van de moedertaal op T2-verwerving, en de visies die T2-verwerving zien als een proces dat lijkt op T1-verwerving. Factoren als leeftijd, intelligentie en dergelijke hangen voor een belangrijk deel samen en beïnvloeden elkaar. Taalcontact en taalaanbod worden als de belangrijkste factoren gezien in het succes van T2-verwerving. De simultane taalverwerving wordt besproken aan de hand van zeven vragen, waaronder de vraag of de beide talen zich al vanaf het begin gescheiden ontwikkelen. Recent onderzoek laat zien dat tweetalig opgroeiende kinderen al vroeg equivalenten kunnen gebruiken, wat pleit voor de zogenaamde Gescheiden Systeem Hypothese (tegenover de oudere Eén Systeem Hypothese).

In hoofdstuk 10, geschreven door Annemarie Schaerlaekens en Sieneke Goorhuis-Brouwer, wordt gesproken over taalproblemen en taalstoornissen. De auteurs gaan in op spraakstoornissen (stem- en articulatiestoornissen en stotteren) en taalstoornissen (specifieke taalstoornissen, zoals afasie, en niet-taalspecifieke stoornissen, die kunnen optreden als gevolg van bijvoorbeeld slechthorendheid en emotionele stoornissen).

In het elfde en laatste hoofdstuk gaat Paul van Geert eerst in op het taalverwervingsprobleem (het leren van een eindige verzameling regels om een oneindig aantal taaluitingen te kunnen produceren), de soorten verwervingsmechanismen en taalverwervingsfactoren, waarna hij een overzicht geeft van de theorieën over taalverwerving.

Er valt over dit werk veel positiefs te melden. Het is het meest uitgebreide handboek over kindertaalverwerving van het Nederlands. Het is zeer up-to-date en geschreven door een keur van onderzoekers die kunnen bogen op een respectabele expertise. Bovendien hebben de redacteuren het voor elkaar gekregen een grote coherentie tot stand te brengen in de presentatie. Over de hele linie is het boek zeer helder geschreven, zijn vaktermen zo veel mogelijk uitgelegd en is de informatie duidelijk gestructureerd. De toegankelijkheid is nog vergroot

doordat een uitgebreid zaken- en personenregister zijn toegevoegd. Natuurlijk is er altijd wel iets aan te merken en daarop vormt dit boek geen uitzondering. Ik beperk mij tot enkele puntjes. Zo valt het laatste hoofdstuk uit de toon, doordat de auteur in mijn ogen te veel informatie in zijn tekst heeft willen stoppen, waardoor het hoofdstuk nogal langdradig is geworden. In het hoofdstuk over de fonologische ontwikkeling wordt me niet duidelijk hoe het nu precies zit met de structuur van het rijm. Zo wordt bijvoorbeeld gesteld dat het rijm tweeplaatsig is, maar tegelijk wordt gezegd dat een lange vocaal (die twee posities binnen de nucleus inneemt) gevolgd kan worden door een obstruent in de coda. Nu wordt als oplossing gekozen voor een templaat met extrasyllabische consonanten, maar dan is het me met de gegeven structuren nog niet duidelijk wat zich nu precies waar in het rijmtemplaat bevindt. Ten slotte dan nog een opmerking over de pragmatiek. Er wordt gesteld dat bepaalde taalhandelingen universeel zijn. Verderop wordt opgemerkt dat de volgorde in de verwerving van taalhandelingen taalspecifiek is. Het is echter niet duidelijk hoe dergelijke verschillen iets met taalspecifieke kenmerken te maken kunnen hebben. Hier rijst dan ook de vraag in hoeverre we bij de gesignaleerde verschillen de oorzaak wellicht moeten zoeken in de verschillende culturen. Taalhandelingen zijn immers onderdeel van de taalgebruikssystematiek en als zodanig ingebed in de cultuur van de gemeenschap waarin een taal wordt gesproken.

*Jan Nijen Twilhaar*

# Signalementen

***Linguistics in the Netherlands 2000***

In deze bundel is een deel van de lezingen samengebracht die werden gehouden op de 31ste jaarlijkse bijeenkomst van de Algemene Vereniging voor Taalwetenschap te Utrecht op 5 februari 2000. Van de negentien artikelen noemen we alleen die met de meest directe relevantie voor deze rubriek. Bert Botma en Erik Jan van der Torre geven *The prosodic interpretation of sonorants in Dutch*. Johanneke Caspers is *Looking for melodic turn-holding configurations in Dutch*. Leonie Cornips onderzocht *The use of* gaan + *infinitive in narratives of older bilingual children of Moroccan and Turkish descent*. Ben Hermans en Marc van Oostendorp bekeken *Voice-tone interaction in a Limburg dialect: Evidence for feature licensing*. Vincent K. van Heuven en Judith Haan hielden zich bezig met de vraag *When and how do we hear whether a Dutch speech utterance is a statement or a declarative question?* Heleen Hoekstra ontwierp *An algorithm for the assignment of sentence accents in Dutch*. Jarich Hoekstra boog zich over *The West Frisian quantifier system and the "mass only" puzzle*. Bart Hollebrandse bekeek de *State-non state difference in Dutch L2 acquisition of English*. Oele Koornwinder en Henk Verkuyl lichten ons in over *Morphological effects of lexical aspect*. Jan Koster en Jan-Wouter Zwart schrijven iets over de *Transitive expletive constructions and the object shift parameter*. Francine Swets verdiepte zich in de *Tilburg Dutch vowel alternations and phonological elements*. Hans Van de Velde en Roeland van Hout onderzochten *N-deletion in reading style*. De bijdrage van Mark de Vries gaat over *Appositive relative clauses* en Ton van der Wouden ten slotte weet ons iets te melden over *Focus on appendices in Dutch*.

**Bibliografische gegevens:**

Hoop, H. de & T. van der Wouden (red.), *Linguistics in the Netherlands 2000.* Amsterdam/Philadelphia: John Benjamins, 2000. 244 blz.

*Jan Nijen Twilhaar*

***Studies op het gebied van de dialecten***

Hieronder komen drie studies op het gebied van de dialecten voor het voetlicht. De eerste is een synchrone klank- en vormleer, de tweede gaat over dialectverandering en de derde studie is een indeling van de Nederlandse streektalen volgens de FFM.

In 2000 verscheen van de hand van J.J. Spa *De dialecten van Groot-IJsselmuiden. Klank- en vormleer*. Het boek beschrijft de klank- en vormleer van de dialecten van Groot-IJsselmuiden, waaronder we moeten verstaan de gemeente IJsselmuiden ('s-Heerenbroek, Wilsum, IJsselmuiden en Zalk). De beschrijving vindt plaats aan de hand van gegevens die in de jaren 1978-1981 door de IJsselacademie werden verzameld met een grote schriftelijke enquête. Als we dit werk louter beoordelen op grond van het het doel waarmee het geschreven is (het tamelijk gedetailleerd vastleggen van de klank- en vormverschijnselen in de desbetreffende dialecten), dan verdient dit boek veel lof, doordat de feiten uitvoerig en op een heldere, prettig te lezen manier worden beschreven. Interessant is verder dat de auteur bij de bespreking van de

klanken uitgebreid ingaat op de historische achtergronden. Een beschrijving van taalfeiten zoals deze, waarbij het taaltheoretische kader ontbreekt, kan echter allerlei vragen oproepen. Een aantal taalverschijnselen krijgt namelijk een nogal toevallig karakter.

In het proefschrift *Structurele en sociale aspecten van dialectverandering* van Reinhild Vandekerckhove wordt de dynamiek van het Deerlijkse dialect onderzocht. Deze uitvoerige studie probeert inzicht te geven in dialectresistentie en dialectverandering. Het werk is opgebouwd uit vier delen: I. Inleiding, met daarin een toelichting op het uitgangspunt, de opzet van het onderzoek en de demografische en economische aspecten van het onderzoeksgebied, II. De analyse van de linguïstische variabelen in het Deerlijkse dialect, III. Verklaringen en interpretatiemodellen, en IV. De West-Vlaamse taalsituatie: taalverhoudingen in West-Vlaanderen. De auteur komt op grond van zijn studie tot drie conclusies. In de eerste plaats is het Deerlijks een West-Vlaams dialect. In de tweede plaats lijkt het dialect meer op het Standaardnederlands dan vroeger. In de derde plaats wijkt het dialect nog voldoende van het Standaardnederlands af om voor een dialect te mogen doorgaan.

In *De indeling van de Nederlandse streektalen* klasseren Cor en Geer Hoppenbrouwers 156 steden en dorpen volgens de FFM. Dit is de afkorting van featurefrequentiemethode, waarbij de computer de fonetische kenmerken telt in teksten uit de *Reeks Nederlandse Dialectatlassen*, waarbij wordt vastgesteld in welke mate deze kenmerken of features voorkomen. Het boek is bedoeld voor een groot publiek en de auteurs hebben dan ook gestreefd naar optimale toegankelijkheid door taalkundige begrippen ter plaatse toe te lichten en het gebruik van fonetisch schrift te beperken.

**Bibliografische gegevens:**

Spa, J.J., *De dialecten van Groot-IJsselmuiden. Klank- en vormleer*. Kampen: Stichting IJsselacademie, 2000. 148 blz. ISBN 90-6697-115-0. NLG 29,95.

Vandekerckhove, R., *Structurele en sociale aspecten van dialectverandering. De dynamiek van het Deerlijkse dialect*. Gent: KANTL, 2000. 352 blz. ISBN 90 72474 26 0. EUR 21,00.

Hoppenbrouwers, C. en G., *De indeling van de Nederlandse streektalen. Dialecten van 156 steden en dorpen geklasseerd volgens de FFM.* Assen: Van Gorcum, 2001. VIII + 212 blz. ISBN 90 232 3731 5. EUR 22,67.

*Jan Nijen Twilhaar*

***Over de betekenis van woorden***

Bij de Sdu verschenen twee boeken over de betekenis van woorden. In *Guichelheil*, dat werd samengesteld door de Taaladviesdienst Genootschap Onze Taal, worden vragen beantwoord over woordbetekenis en over de herkomst van woorden. In dit boek, dat een inleiding bevat van Riemer Reinsma, heeft deze adviesdienst de leukste vragen bijeengebracht. De woorden zijn alfabetisch gerangschikt en verwerkt in een vraag. Daaronder staat het antwoord van de adviesdienst. Aan het eind van het boek staat een woordregister.

Ook *Woorden en hun betekenis* van Ton den Boon gaat, anders dan de titel van het boek doet vermoeden, niet alleen over betekenis, maar ook over de geschiedenis van woorden. Het boek bestaat uit vijf delen, waarin de volgende onderwerpen aan de orde komen: Woorden en hun betekenis, Betekenisverandering, De gevoelswaarde van het woord, Ambiguïteit, en Betekenisbeschrijving.

**Bibliografische gegevens:**

Boon, T. den, *Woorden en hun betekenis.* Den Haag: Sdu, 2001. 171 blz. ISBN 90 12 08961 1. NLG 25,00.

*Jan Nijen Twilhaar*

# Uit de tijdschriften

De rubriek *Uit de tijdschriften* geeft kort weer wat er in andere tijdschriften op het gebied van de taalkunde is verschenen. Momenteel worden in deze rubriek de volgende tijdschriften besproken: *Anéla, Driemaandelijkse Bladen, Gramma/TTT, Leuvense Bijdragen, Naamkunde, Nederlands van Nu, Neerlandica Extra Muros, Ons Erfdeel, Onze Taal, Spiegel, Southern African Linguistics and Applied Language Studies, Taal en Tongval, Taalkundig Bulletin, Tijdschrift voor Nederlandse Taal- en Letterkunde, vakTaal, VDW-berichten, De Woordenaar*. Uitgevers van niet vermelde taalkundige periodieken die hun tijdschrift besproken willen zien, wordt verzocht contact op te nemen met de redacteur van deze rubriek:

Dr. J. Nijen Twilhaar
Oerdijk 35
7433 AG Schalkhaar
tel.: 0570-608080
e-mail: jnt@xs4all.nl

**Nederlands van Nu**
50 (2002), nr. 3

Dit derde nummer begint met een bijdrijge van Filip Devos over nationaliteitsnamen in vaste uitdrukkingen: *Gaat u wel eens Nederlands?* Siegfried Theissen neemt in *Schrijft Knack Belgisch? (7)* opnieuw het taalgebruik van het tijdschrift *Knack* onder de loep. Isabelle 's Heeren verdiepte zich in *De verbuiging van het adjectief na een bezittelijk voornaamwoord voor een onzijdig enkelvoudig zelfstandig naamwoord*. Peter Debrabandere geeft zijn derde Schooltaaltip, die deze keer gaat over cursussen, syllabi, leerboeken en handboeken. Deze bijdrage wordt gevolgd door een artikel van dezelfde auteur over het woord *Schendeventen(n)*. In *Spelen met taal* geeft Gilbert De Bruycker opnieuw ideeën voor het basis- en het voortgezet onderwijs. Marc De Coster bespreekt in *Nieuwspraak* een aantal neologismen.

Verder zijn er de kleinere bijdragen en de vaste rubrieken.

**Ons Erfdeel**
45 (2002), nr. 3

Het derde nummer van deze jaargang bevat onder meer een artikel van Kees Groeneboer, waarin wordt uitgelegd *Waarom het Nederlands geen wereldtaal is geworden*.

**vakTaal**
15 (2002), nr. 1

In het eerste nummer van deze jaargang vinden we onder meer een bijdrage van Wim Klooster: *Grammatica voor de liefhebber IV*. Marc van Oostendorp geeft in zijn artikel *Taal om bij te dansen* zijn visie op het gebruik van leenwoorden. In *Grammatica voor mopperpotten* geeft Louise Cornelis een bespreking van Wim Kloosters boek *Grammatica van het hedendaags Nederlands*. Verder is er een reactie van A.M. Duinhoven op de bespreking van zijn boek in het voorgaande nummer van dit tijdschrift.

# Ontvangen boeken

**Abeling, André.** *Het groene woordenboek.* Den Haag: Sdu, 2002. 901 blz. ISBN 90 12 09308 2. EUR 25,00.

**Bierce, Ambrose.** *Het duivels woordenboek*. Amsterdam: Contact, 2002. 112 blz. ISBN 90 254 1226 2. EUR 12,90.

**Broek, M.A. van den.** *Erotisch spreekwoordenboek*. Amsterdam: Veen, 2002. 127 blz. ISBN 90 204 0003 7. EUR 9,95.

**Janssen, Theo (red.).** *Taal in gebruik. Een inleiding in de taalwetenschap.* Den Haag: Sdu, 2002. 282 blz. ISBN 90 12 09483 6. EUR 22,00.

**Leezenberg, Michiel & Gerard de Vries.** *Wetenschapsfilosofie voor geesteswetenschappen*. Amsterdam: AUP, 2001. 256 blz. ISBN 90 5356 465 9. EUR 18,50.

**Muniz, Gisa.** *Portugees voor zelfstudie*. Utrecht: Het Spectrum, 2002. 360 blz. ISBN 90 274 6985 7. EUR 25,00.

**Oostendorp, Marc.** *Steenkolen-Engels. Een pleidooi voor normvervaging*. Amsterdam: Veen, 2002. 159 blz. ISBN 90 204 5749 7. EUR 13,50.

**Spellingwijzer Onze Taal.** Amsterdam: Veen, 2002. 646 blz. ISBN 90 204 0114 9. EUR 13,50.

**Toorn-Schutte, Jennie van der.** *Cultuur en tweedetaalverwerving. Een taalkundig-antropologische vergelijking tussen Oost en West.* Amsterdam: Boom, 2001. 190 blz. ISBN 90 5352 605 6. EUR 18,90.

**RECTIFICATIE**

Geheel buiten de schuld van de auteur is er in de productiefase van het vorige nummer, 7.3, een fout geslopen in de paragraafnummering van het artikel van Frank Joosten, 'De uitspraak van letterwoorden in het Nederlands' ( p. 238-263). Daardoor geven de paragraafnummers in de verwijzingen binnen de tekst van het artikel steeds één nummer te laag aan. Men leze dus 'zie 2.1' als 'zie 3.1' en 'zie 3.2' als 'zie 4.2', enz.