

Pragmatische features in DenK

Citation for published version (APA):

Beun, R. J., & Piwek, P. L. A. (1997). *Pragmatische features in DenK: PRAGTAGS D(4,9)-96-06-spec.* (IPO rapport; Vol. 1147). Instituut voor Perceptie Onderzoek (IPO).

Document status and date:

Gepubliceerd: 20/02/1997

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:


openaccess@tue.nl

providing details and we will investigate your claim.

Rapport no. 1147

**Pragmatische features in
DenK: PRAGTAGS**

**Robbert-Jan Beun
Paul Piwek**



Voor akkoord: Dr. ir. J.H. Eggen

Pragmatische features in DENK: PRAGTAGS
D(4,9)-96-06-spec

Robbert-Jan Beun & Paul Piwek

Inhoudsopgave

1	Introductie	2
2	Achtergrond	2
3	Waarheidsconditionele Ambigüiteit	5
3.1	Referentiële ambigüiteit	6
3.2	Overige ambigüiteiten	7
4	Temporele Informatie	9
5	Functionele Interpretatie en Reactie	10
5.1	Het vaststellen van de taalhandeling	11
5.2	Het gebruik van speciale werkwoorden	15
5.3	Het modale hulpwerkwoord can	16
6	Prominentie en WH-vragen	20
7	Overige features	23
8	Conclusies	24
9	Referenties	25
10	Appendix: Overzicht pragmatische features	27

1 Introductie

Informatie in natuurlijke taaluitingen kan verdeeld worden in twee types informatie: a. informatie die bijdraagt aan de *waarheidsconditionele* aspecten van de uiting met betrekking tot een specifiek gespreksdomein en b. informatie die bijdraagt aan de *functionele* aspecten van de uiting in termen van de verrichte taalhandeling en de te verwachten reactie. Zo bevat de uiting

(1.1) 'Is the electron beam directed to this plane?'

enerzijds waarheidsconditionele informatie, zoals het bestaan van een 'electron beam' en een specifiek 'plane', en anderzijds functionele informatie die bijvoorbeeld de vraagfunctie van de uiting aangeeft. In natuurlijke dialogen kunnen beide types informatie in het algemeen niet volledig geïnterpreteerd worden zonder gebruikmaking van contextuele informatie.

In dit verslag richten we ons op de formele representatie in het DENK-systeem van een taaluiting vlak voordat contextuele aspecten een rol spelen in het interpretatieproces. In deze representatie wordt ruwweg onderscheid gemaakt tussen enerzijds de waarheidsconditionele aspecten van de uiting en anderzijds de functionele en contextuele – ook wel de *pragmatische* – aspecten van de uiting. We richten ons hier met name op het pragmatische deel van de uiting.

De pragmatische aspecten worden uitgedrukt in een aantal zogenaamde (*pragmatische*) *features* met hun bijbehorende waarden. In dit rapport geven we naast een overzicht van de features die een rol spelen bij het interpretatie- en dialooggedrag van de assistent ook een argumentatie voor de opname van een bepaald type annotatie. Hierbij zullen in sommige gevallen ook empirische data ondersteuning geven aan onze beslissingen. Vooruitlopend op de module Conduct laten we in sommige gevallen regels zien die het gedrag van de assistent bepalen aan de hand van de aanwezige features. Het precieze algoritme hiervoor zal echter beschreven worden in de deliverable Conduct.

2 Achtergrond

Voordat we een formele definitie geven van de hier weergegeven pragmatische features maken we eerst enkele aannames expliciet die in de communicatieve situatie en de domeinkeuze van DENK zijn opgesloten. Deze aannames hebben namelijk gevolgen voor het soort pragmatische features dat opgenomen zal worden en de gedragsregels die hierop van toepassing zijn.

- Er zijn slechts twee dialoogdeelnemers, de assistent en de gebruiker. Het is daardoor altijd duidelijk tegen wie een uiting is gericht. In een domein waarbij verschillende assistenten, en dus ook verschillende contexten aanwezig zijn, zou de mogelijkheid moeten bestaan een uiting naar een specifieke interpreteerder te zenden (bijv. 'Paul, ruim jij de rommel even op!'). Verder spelen zowel geslacht (*gend*), als getal (*num*) en persoon (*pers*) nog geen rol in het interpretatieproces, omdat we (nog) geen meervouden behandelen en omdat er geen personen voorkomen in het gespreksdomein.

- De assistent is alwetend over het gespreksdomein en gedraagt zich bovendien als een expert over het domein. De assistent zal dus geen informatie over het domein die geuit wordt door de gebruiker toevoegen als nieuwe kennis aan de reeds bestaande privé-kennis (dit geldt uiteraard niet voor de *gemeenschappelijke* achtergrondkennis (Common Ground) die wel tijdens de dialoog verandert); declaratieven over het domein worden derhalve als vragen geïnterpreteerd.
- In het DENK-systeem zijn we door het onderscheid contingente/universele domeinkennis in staat 'fysische' wetten te representeren en hierover te redeneren. We hebben op dit moment echter niet de mogelijkheid sociale of institutionele wetten te representeren, zoals die nodig zouden kunnen zijn in een verkeers- of juristendomein. Modaliteiten zoals 'must' en, in sommige gevallen, 'can' zullen daarom behandeld worden als deel van de indirecte taalhandeling.
- Functionele informatie zal, in tegenstelling tot pure domein-informatie, contextonafhankelijk worden behandeld. We denken dat dit voorlopig een werkbaar principe is, maar uiteindelijk zal ook bij de interpretatie van deze informatie de context een rol spelen. Op dit moment speelt dit geen rol omdat de assistent alwetend is over het domein en zich als een expert gedraagt. Indien de assistent tevens eigen intenties heeft en zijn alwetendheid verliest (en zich hiernaar gedraagt) zal de interpretatie waarschijnlijk veel contextgevoeliger zijn.¹

We zullen hieronder niet opnieuw het totale interpretatieproces beschrijven, maar alleen datgene wat van belang is om dit rapport te kunnen begrijpen. Voor uitgebreidere informatie verwijzen we naar bijvoorbeeld Ahn et al. (1995), Piwek (1995), Verlinden (1996) en Kievit (1996).

Tot nu toe werd in DENK de semantische representatie van een uiting op twee niveaus onderscheiden (Ahn, et al. 1995). De eerste representatie, ULF genaamd (Kievit, 1996), is het resultaat van een structurele analyse van een uiting van de gebruiker, het zogenaamde parseerproces. Omdat in de ULF-expressie nog geen contextuele informatie is betrokken, is deze representatie in het algemeen ondergespecificeerd ten opzichte van de tweede representatie die in Typentheorie wordt weergegeven. Bovendien bevat de ULF-expressie, behalve waarheidsconditionele informatie, ook informatie over de functie van de uiting.

Bij het omzetten van ULF naar CTT maken we gebruik van een tussenliggende representatie, de zogenaamde *geannoteerde segmenten*. De geannoteerde segmenten bestaan uit paren van Typentheoretische expressies met bijbehorende annotaties en worden, vóór toevoeging aan de Common Ground, op de Pending Context² geplaatst (Piwek, 1995), waar respectievelijk de modules Clues en Conduct het verdere interpretatie- en reactieproces verzorgen. De annotaties bevatten de al eerder genoemde pragmatische features met hun bijbehorende waarden (of 'values'). We herhalen hier voor de volledigheid de definities zoals gegeven in Piwek (1995):

¹De functionele interpretatie van een declaratief kan dan bijvoorbeeld afhangen van het onderwerp waarover wordt gepraat.

²Deze context bevat representaties van uitingen die nog niet volledig zijn afgehandeld. Bijvoorbeeld een vraag die nog niet is beantwoord.

Def 1. (feature-value) Een feature-value paar is een paar <A,B>, waarbij A een (pragmatisch) feature is en B zijn waarde.

Def 2. (annotatie) Een annotatie is een verzameling feature-value paren.

Def 3. (geannoteerd segment) Een geannoteerd segment is een paar <S,A>, waarbij S een (negatief) segment is en A een annotatie.

Hieronder zullen we voor de duidelijkheid feature-value paren weergeven als A(B).³ De waarde(n) die een feature kan aannemen is afhankelijk van het soort feature.

De features met hun bijbehorende waarden representeren een grote verscheidenheid aan informatie die, naast de bijdrage aan de contextafhankelijke waarheidsconditionele aspecten en de functie van de uiting, ook informatie bevat over het verwerkingsproces zelf. In feite kunnen we de features in vier categorieën verdelen:

1. Features die gebruikt worden om de *waarheidsconditionele inhoud* van de uiting te *berekenen*, zoals het feature dat aangeeft dat een specifieke marker gebonden moet worden aan de context (RB: Requires Binding) of het feature dat aangeeft welk type definitief is gebruikt in de verwijzing (MT: Marker Type). Zodra de markers zijn gebonden spelen de features geen rol meer in de representatie. Een uitzondering hierop vormt het feature TEMP_TAGS dat wordt gebruikt om de temporele informatie van het segment weer te geven en dat niet komt te vervallen na het interpretatieproces.
2. Features voor het representeren van de *functionele informatie* van de uiting. Hier maken we onderscheid tussen de features die direct uit de ULF representatie volgen, zoals punctuatie (PUNC) en de imperatieve modus van de VP (IMP), en de features die uit de 'directe' features kunnen worden berekend, namelijk een vraag (RI: Request for Information), een verzoek (RA: Request for Action) of een vraag naar mogelijkheden (RI_POS, bijv. 'Can the em be in HM-mode?').
3. Features voor het opslaan van *relaties tussen de verschillende uitingen* op de Pending Context (bijv. de vraag-antwoord relatie). Deze features worden in dit rapport niet behandeld, omdat ze nauw verweven zijn met het management van de Pending Context door Conduct; maar zie Piwek (1997).
4. Features die informatie geven over de *status van het verwerkingsproces* van uitingen. Hiertoe rekenen we onder andere het feature UO (Unresolved Object) dat aangeeft in hoeverre een uiting reeds verwerkt is of dat bepaalde type restricties op een object van toepassing zijn (TR: Type Restriction).

Een belangrijk verschil tussen de 'waarheidsconditionele' features uit de eerste categorie en de 'functionele' features uit de tweede categorie is dat de informatie uit de directe functionele features in sommige gevallen weer gebruikt kan worden in de reactie naar de gebruiker. We zullen hier later nog uitvoerig op terugkomen, maar geven nu vast een indruk van dit proces. Stel we hebben de directe vraag:

³Op andere plaatsen wordt ook wel de notatie A=B gebruikt.

(2.1) 'Can you tell me whether the lens is on?'

Deze vraag wordt dikwijls indirect opgevat als:

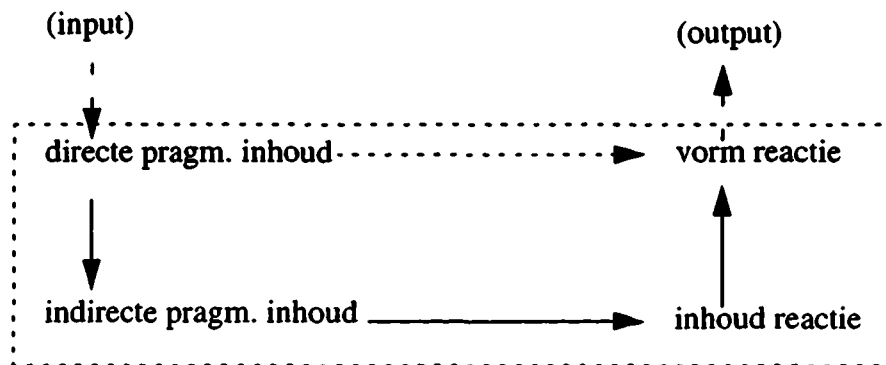
(2.2) 'Is the lens on?'

wat bijvoorbeeld correspondeert met het volgende (versimpelde) geannoteerde segment:

(2.3) <[X:on-lens318], [RI(X), ...]>

waarbij RI(X) aangeeft dat de hoorder moet communiceren of er een bewijs is voor de propositie on-lens318. Maar als we de vraag volledig zouden reduceren tot (2.3), zou dit problemen kunnen opleveren met het geven van een adequaat antwoord. Immers, enkel 'Yes' zou hier bijvoorbeeld kunnen betekenen 'Yes, I can tell you the answer' of 'Yes, the lens is on', terwijl 'No' kan betekenen 'No, I cannot tell you whether the lens is on' of 'No, the lens is not on'.

Het onderscheid tussen directe en indirecte pragmatische functie speelt een belangrijke rol bij het verwerkingsmodel dat we hanteren. Het model ziet er ruwweg als volgt uit: aan een uiting wordt zowel een directe als indirecte functie toegekend. Op grond van de indirecte functie wordt de inhoud van een reactie berekend. Vervolgens kan de directe betekenis gebruikt worden om de vorm van de reactie te bepalen. We hebben dit proces geschetst in Figuur 1 (zie verder ook sectie 5).



Figuur 1: Model van de verwerking van indirecte taalhandelingen.

3 Waarheidsconditionele Ambigüiteit

Een belangrijk deel van de betekenis van de uiting kan veelal niet opgelost worden op basis van alleen lexicaal en syntactische informatie. De geannoteerde segmenten kunnen dan ook, onder andere afhankelijk van de voortgang van het interpretatieproces, een grote mate van ondergespecificeerdheid bezitten t.a.v. de CTT-representaties in de cognitieve toestand van de assistent. Deze semantische ambigüiteiten dienen dan, voor zover mogelijk, met behulp van de reeds aanwezige contextuele informatie te worden opgelost; een eenvoudig voorbeeld daarbij is:

(3.1) A: Waar werkt Piet?

B: Piet zit op de bank.

waarbij het woord 'zit' wordt gedesambigueerd door de voorafgaande context (en daarmee dus ook 'bank'). Als de betekenis van een woord niet door de context opgelost kan worden, kan in principe altijd een wedervraag gesteld worden aan de gebruiker. We zullen hier niet bekijken hoe de verschillende types ambigüiteiten opgelost kunnen worden,⁴ maar slechts een overzicht geven van enkele pragmatische features die, voor zover op dit moment bekend, noodzakelijk zijn binnen het geannoteerde segment om ambigüiteiten m.b.v. de context op te lossen.

In Kievit (1996) worden de volgende soorten ambigüiteit onderscheiden: *referentiele* (pronomina, demonstratieven, etc.), *lexicale vaagheid* (groot, zwaar, etc.), *lexicale ambigüiteit* (bank, pas, etc), *structureel syntactisch* ('De vrouw sloeg de man met de onderbroek'), *structureel semantisch* ('Every button controls a lens') (zie bijv. ook Bunt, 1995). We hebben er voorlopig voor gekozen vaagheid buiten beschouwing te laten.⁵

3.1 Referentiële ambigüiteit

Referentiële ambigüiteit betreft hier vooral het vinden van de juiste referent bij definiete expressies ('this lens', 'the button', etc.). Hoe dit proces in zijn werk gaat wordt uitvoerig beschreven in Beun & Kievit (1995). Naast de gebruikelijke semantische restricties wordt hierbij ook rekening gehouden met het type definit, hieronder aangeduid door *def_type*, dat richting geeft aan het zoekproces naar het correcte antecedent. De waarde van *def_type* (bijvoorbeeld pronomina, demonstratief, etc.) wordt samen met de marker die verwijst naar de bewoner van het nog te resolveren object ondergebracht in het feature MT (Marker Type). MT bevat voor iedere uiting dus een lijst van paren bestaande uit een marker met de bijbehorende waarde van het type definit.

Alle markers die een binding vereisen vanuit de context worden ondergebracht in het feature RB (Requires Binding). In deze representatie wordt tevens rekening gehouden met mogelijke relata (zoals 'the lens below the specimen'), deze moeten immers opgelost worden voordat het 'target' object opgelost kan worden. Bij iedere 'definiete' marker zijn bovendien markers toegevoegd die refereren naar de bewoners van de statements die de eigenschappen van het te resolveren object aangeven en daarmee dus tevens de semantische restricties aangeven op het object.⁶ RB bevat derhalve een lijst van marker-lijsten (*M_list*). In de annotatie wordt tevens met behulp van het feature UO (Unresolved Object) bijgehouden welke verwijzing geen succesvol resolveringsresultaat oplevert. UO heeft twee argumenten: het eerste geeft

⁴Een van de problemen is dat het oplossen van ambigüiteiten afhankelijk is van de doelen van de gesprekspartners. Vergelijk bijv. 'Staat het gele blok naast het rode?' met 'Plaats het gele blok naast het rode!'; in het tweede geval moet de assistent uiteindelijk altijd een keuze maken over de precieze locatie van het gele blok, in het eerste geval doet dit er niet toe zolang het maar naast het rode staat.

⁵In hoeverre de verschillende types ambigüiteiten een rol spelen in de praktijk is op dit moment een open vraag en dient nog onderzocht te worden. In tegenstelling tot bijvoorbeeld de functionele informatie is het moeilijk dit onderzoek te baseren op de bestaande dialogen; voorlopig proberen we zo veel mogelijk uit te gaan van onze intuïties. Later kan een en ander getest worden met het prototype (zie echter Verbeem. 1996).

⁶Deze eigenschappen worden in feite direct afgeleid uit de 'scoop' van de quasi-determiner in de ULF-representatie.

aan welke verwijzing heeft gefaald, het tweede, dat bestaat uit een lijst van objecten, geeft aan welke objecten mogelijkwijs in aanmerking komen voor substitutie. Indien het eerste argument wel een verwijzing bevat, maar de lijst van het tweede argument leeg is, heeft Resolve geen passende referent kunnen vinden en heeft het proces dus gefaald. Indien na afloop van het resolveringsproces beide argumenten leeg zijn is het proces geslaagd.

- $RB(list_of_M_list)$

$list_of_M_list \rightsquigarrow [M_list \mid list_of_M_list] \mid []$

$M_list \rightsquigarrow [marker \mid M_list] \mid []$

- $MT(def_list)$

$def_list \rightsquigarrow [(marker, def_type) \mid def_list] \mid []$

$def_type \rightsquigarrow def_art \mid subst_demon \mid attr_demon \mid nom_pronoun \mid one_indef$
 $\mid one_def \mid one_demon$

- $UO(marker, O_list)$

$O_list \rightsquigarrow [object \mid O_list] \mid []$

Dit maakt het mogelijk uitingen te behandelen als:

(3.2) ‘Increase the magnification at this plane!’

$\langle [P:plane, M:magnification, A:at-M-P, \dots],$

$[RB([P], [M, A]), MT((P, attr_demon), (M, def_art)), \dots] \rangle$

Merk op dat via de marker A in het tweede element van RB impliciet de verwijzing naar het relatum ‘this plane’ opgesloten zit.

3.2 Overige ambiguïteiten

In feite kunnen alle overige vormen van ambiguïteit (behalve vaagheid) uitgedrukt worden als een soort disjunctie. Deze disjunctie bevindt zich echter niet op het niveau van intentioneel overgedragen informatie, maar zou ruwweg zoiets kunnen betekenen als: ‘de uiting betekent p’ of ‘de uiting betekent q’ (en dus niet ‘de uiting betekent p of q’!). In (3.1) hebben we dan bijvoorbeeld twee lezingen voor ‘zit’ en twee lezingen voor ‘bank’, wat in totaal vier disjuncten op zou kunnen leveren.⁷

Binnen DENK worden deze disjuncties gerepresenteerd op het niveau van het geannoteerde segment, wat inhoudt dat de disjunctie opgelost moet worden voordat een van de disjuncten toegevoegd kan worden aan de Common Ground. Deze disjunctie bevindt zich dus op het ‘proces-niveau’. In Kievit (1997) wordt daarvoor gebruik gemaakt van een feature TR (Type Restriction), waarin restricties worden aangebracht op de types van de variabelen die voorkomen in het CTT-deel van het geannoteerde segment. In feite wordt het feature TR dus gebruikt om meta-variabelen over types te kunnen invoeren en de bijbehorende informatie geeft daarbij het bereik aan van deze variabelen in termen van de toegestane types. De disjunctie (disj) geeft hierbij dan aan tussen welke types uiteindelijk een keuze gemaakt kan worden.

⁷Of dat ook werkelijk zo is hangt weer af van de combinaties. Zo zouden we op voorhand al de lezing kunnen uitsluiten dat Piet zich bovenop een financiële instelling bevindt, alhoewel dat laatste natuurlijk niet onmogelijk is.

Naast het gebruik van de disjunctie in het TR-feature is het ook mogelijk andere soorten restricties op te leggen met dit feature. Dit gebeurt bijvoorbeeld wanneer vanuit het conceptueel lexicon niet genoeg restricties opgelegd kunnen worden, zoals bij het gebruik van pronomina. Zo kan in:

(3.3) ‘It reaches the plane.’

het object waar door middel van het pronomen ‘It’ naar verwezen wordt van het type `bundle` zijn, maar door inheritance kan het ook van een type zijn dat lager staat in de hiërarchie (hieronder `leq`).

De definitie van het TR-feature ziet er dan als volgt uit:

• $TR(list_of_TypRes)$

$list_of_TypRes \rightsquigarrow [(marker:TypeRestriction) | list_of_TypRes] | []$

$TypeRestriction \rightsquigarrow disj(list_of_indexen, list_of_Vtype) | leq(Vtype) | geq(Vtype)$

$list_of_indexen \rightsquigarrow [index | list_of_indexen] | []$

$list_of_Vtype \rightsquigarrow [Vtype | list_of_Vtype] | []$

$Vtype \rightsquigarrow marker | type$

waarbij *type* een CTT-type is en de *list_of_indexen* de verschillende mogelijkheden verbinden met andere disjuncties in de annotatie. We zullen hier niet verder op het desambigueringsproces ingaan, maar verder verwijzen naar Kievit (1997). Wel kan nog toegevoegd worden dat indien een bepaalde index unificeert met `no`, de mogelijkheid komt te vervallen en de disjunctie reduceert naar een eenvoudigere term; in het gunstigste geval verdwijnt de hele disjunctie. Rest ons nog een tweetal voorbeelden:

(3.4) ‘It reaches the plane.’

```
<[P:plane, A:*, B:A, D:reaches-B-P],
  [RB([P],[B,A]), MT((P,def_art),(B,nom_pronoun)),
  TR(A:leq(bundle)),...]>
```

In (3.4) wordt dus door middel van het TR-feature aangegeven dat de marker A (het type van B) van het type `bundle` is of lager.

(3.5) ‘The bundle reaches it.’

```
<[A:*, C:A, B:bundle, D:E],
  [RB([C,A],[B]), MT((C,nom_pronoun),(B,def_art)),
  TR(A:leq(F), F:geq(A),
  E:disj([G,H],[reaches-B-C, reaches-B-(p_of-C)]),
  F:disj([G,H],[plane,axobj])),...]>
```

We zien dat in (3.5) sprake is van zowel een ambiguïteit in het werkwoord ‘reach’ als in het bijbehorende argument vanwege het gebruik van het pronomen. vandaar de dubbele (afhankelijke) disjunctie. Na het oplossen van het pronomen met behulp van Resolve kan dan hopelijk een keus gemaakt worden.

4 Temporele Informatie

Omdat de assistent vragen van de gebruiker moet kunnen beantwoorden over temporele verschijnselen binnen de microscoop zal deze informatie ook gerepresenteerd dienen te worden binnen het geannoteerde segment. Er is om praktische redenen voor gekozen deze informatie te vermelden in het geannoteerde deel van het segment.

Een oppervlakkige beschouwing van de tot onze beschikking staande dialogen heeft geleerd dat de in Chronicle (Ahn & Borghuis, 1996) behandelde concepten semantisch gezien ruimschoots de temporele informatie afdekken van de in de dialogen voorkomende expressies.⁸ We sluiten voor wat betreft de temporele informatie in het fragment daarom nauw aan bij datgene wat vermeld is in Chronicle.

Semantisch gezien worden in Chronicle twee types temporele verschijnselen onderscheiden: a. gebeurtenissen (of *events*) die als instantaan worden beschouwd en geen interne structuur bezitten, en b. toestanden (of *propositions*) die gedurende een bepaald tijdsinterval gelden. Events en propositions (samen *eventualiteiten* genoemd) kunnen door middel van informatie uit het werkwoord gepositioneerd worden ten opzichte van het spreekmoment. We spreken dan over **present** indien een segment geldig is tijdens het spreekmoment en **past** indien het geldigheidsinterval van het segment strikt vooraf gaat aan het spreekmoment. Daarnaast kan met behulp van informatie uit de werkwoorden worden overgebracht dat een eventualiteit al dan niet is afgesloten op een temporeel referentiemoment. We spreken dan respectievelijk over het **perfect** en het **imperfect**. Het referentiemoment kan voor het spreekmoment liggen of er gelijk aan zijn.

Indien het referentiemoment en het spreekmoment niet gelijk zijn moet een andere eventualiteit opgepikt worden om de geldigheid van de uitspraak te kunnen bepalen. Dit gebeurt bijvoorbeeld indien een zin in de voltooid verleden tijd staat (**past_perfect**) of indien de zin samengesteld is met behulp van temporele relaties zoals 'before', 'after', 'since' etc. In het laatstgenoemde geval kan de referentie-eventualiteit in de zin zelf worden gevonden⁹, terwijl in de **past_perfect** de referentie-eventualiteit zich dikwijls in de dialoogcontext zal bevinden of een opvallende gebeurtenis kan zijn binnen het domein. Net zoals in het rapport Chronicle zullen we hier niet ingaan op de details van dit resolveringsproces. Wat echter wel belangrijk is, is dat de interpretatiemodule Clues weet dat voor het bepalen van de geldigheid van het segment zo'n eventualiteit noodzakelijk uit de context opgehaald dient te worden, waarmee het resolveringsproces in grote lijnen dus overeenkomt met het algoritme Resolve voor het oplossen van definiëte expressies (Beun en Kievit, 1995).

In het geannoteerde segment dienen we dus zowel informatie op te nemen over de werkwoordstijd als de ligging ten opzichte van mogelijk andere eventualiteiten. We laten in het segment de informatie over het spreekmoment achterwege, omdat deze in feite altijd beschikbaar is op het moment dat de uiting wordt verwerkt in de

⁸We doen hier geen uitspraak over de verschillende linguïstische formuleringen die gebruikt kunnen worden om 'ongeveer' hetzelfde uit te drukken of over de voorgestelde semantische definities van temporele connectieven. Deze aspecten zouden nader onderzocht moeten worden.

⁹Dit kan worden vergeleken met bijvoorbeeld het oplossen van reflexieve pronomina in Resolve, waarbij het antecedent wordt opgepikt in de zin zelf.

Pending Context.¹⁰ Omdat bij de *past_perfect* geldt dat zowel de eventualiteit, het referentiemoment en het spreekmoment verschillen, gaan we er in de onderstaande representatie vanuit dat alleen de *past_perfect* een extra markeerder bevat voor het referentiemoment; deze markeerder, die dan na een geslaagd resolveringsproces verwijst naar de referentie-eventualiteit, kan dan bijvoorbeeld ingevuld worden vanuit de reeds bestaande context.

In de onderstaande representatie van *TEMP_TAGS* verwijst de eerste temporele marker (*temp_mark*) naar een mogelijk aanwezige referentie-eventualiteit in de zin. Ook in *past_perfect(marker, marker)* verwijst het eerste argument naar de referentie-eventualiteit.

• *TEMP_TAGS(connective, temp_mark, temp_mark)*
connective \sim while | before | until | after | when | since | nil
temp_mark \sim *past_perfect(marker, marker)* | *past_imperfect(marker)*
 | *present_perfect(marker)* | *present_imperfect(marker)* | nil

Dit maakt het in principe mogelijk constructies te behandelen als (we presenteren hier alleen het relevante deel van de annotatie):

(4.1) 'Which aperture controlled the brightness of the image?'
 TEMP_TAGS(nil, nil, past_imperfect(X))

(4.2) 'Had the OA aperture been removed?'
 TEMP_TAGS(nil, nil, past_perfect(X,Y))

(4.3) 'Was the C2-lens excited before the OA-aperture had been removed?'
 TEMP_TAGS(before, past_perfect(X,Y), past_imperfect(P))

Belangrijk is op te merken dat we in deze representatie geen rekening gehouden hebben met quantificaties zoals 'Always when de microscope was in MP-mode, the C2-lens is excited.', waarbij de scope van 'always' de gehele samengestelde zin betreft. Dit zou bijvoorbeeld van belang kunnen zijn voor vragen als 'Is the C2-lens excited when the microscope is in MP-mode?', maar in feite lijkt deze zin weinig anders uit drukken dan een 'gewone' als-dan relatie.

Een laatste opmerking betreft het gebruik van progressives. Deze categorie is hier niet opgenomen omdat het microscoop-domein hiertoe weinig aanleiding geeft: de events in dit domein zijn allemaal instantaan, waardoor zich geen situaties voordoen waar bij een event plaatsvindt 'in' een ander event. Zoals bijvoorbeeld in 'When Mary was crossing the street, she was hit by a truck'.

5 Functionele Interpretatie en Reactie

De in het DENK-fragment mogelijk aanwezige functionele informatie kan herkend worden aan zowel morfologisch/lexicale als syntactische kenmerken van de uiting (zie ook Verlinden 1996). Deze informatie wordt in de geannoteerde segmenten

¹⁰We kunnen tot op zekere hoogte de informatie over het spreekmoment later nog verkrijgen uit de volgorde waarin segmenten toegevoegd zijn aan de dialogcontext.

uiteindelijk omgezet in de ‘taalhandeling’ RI (Request for Information), RI_POS (Request for Possibility) of RA (Request for Action). We vatten deze samen onder het feature FORCE, dat als eerste argument de taalhandeling bevat en als tweede argument een lijst van markers (die in veel gevallen uit slechts één element bestaat). In het geval van een RI dient een substitutie plaats te vinden voor de marker z.d.d. het segment bewijsbaar is in de context. Deze substitutie dient vervolgens gecommuniceerd te worden te worden naar de gebruiker. Een speciaal geval hierbij is de taalhandeling RI_POS, die wordt gebruikt indien de gebruiker vraagt naar de mogelijkheden van een bepaalde propositie (zie ook hieronder). Indien de taalhandeling een RA is dient een actie uitgevoerd te worden in het domein; zodra de actie is uitgevoerd wordt een verse CTT-variabele ingevuld voor de marker en wordt deze geschrapt uit de lijst.

• $\text{FORCE}(f_name, M_list)$
 $f_name \rightsquigarrow \text{RI} \mid \text{RI_POS} \mid \text{RA}$
 $M_list \rightsquigarrow [\text{marker} \mid M_list] \mid []$

We geven hierbij de volgende voorbeelden:

- (5.1) ‘Switch off the lens!’
 $\langle [\text{lens318:lens}, \text{P:switchoff-lens318}], [\text{FORCE}(\text{RA}, [\text{P}]), \dots] \rangle$
- (5.2) ‘Which lens is off?’
 $\langle [\text{X:lens}, \text{P:off-X}], [\text{FORCE}(\text{RI}, [\text{X}]), \dots] \rangle$
- (5.3) ‘Can this lens be switched off?’
 $\langle [\text{lens318:lens}, \text{P:switchoff-lens318}], [\text{FORCE}(\text{RI_POS}, [\text{P}]), \dots] \rangle$

In sommige gevallen is het vaststellen van de taalhandeling (met bijbehorende marker) niet voldoende om een adequate reactie te genereren. Zo is het soms noodzakelijk om behalve het antwoord op de directe vraag extra informatie toe te voegen. We zullen hieronder ingaan op beide aspecten van de verwerking, dat wil zeggen dat we enerzijds bekijken hoe de taalhandeling wordt bepaald, en anderzijds zullen we ingaan in op de informatie die van belang zou kunnen zijn bij het genereren van een specifiek antwoord. In enkele gevallen presenteren we hierbij ook de resultaten van eerder opgenomen dialogodata. Het gaat hierbij om terminaldialogen over vluchtinformatie (Beun, 1987) en over het blokkendomein (Cremers, 1993).

5.1 Het vaststellen van de taalhandeling

Binnen het huidige DENK-fragment worden de volgende indicatoren gebruikt om de taalhandeling vast te stellen: punctuatie (PUNC), inversie van (hulp)werkwoord en subject (INV), de modus van de VP (IMP), WH-informatie (WHVAL), tussen-of achtervoegingen (TAIL), en het gebruik van speciale werkwoorden (VERB_CONS), zoals modale hulpwerkwoorden. We zullen de laatste categorie apart behandelen. We beginnen met een definitie en onderbouwing van de overige categoriën.

Punctuatie De afsluiting van de zin. We accepteren ook zinnen zonder punctuatie (waarde van *clospunctval* is dan nil).

- PUNC(*clospunctval*)
clospunctval \leadsto period | questmark | exclmrk | nil

Inversie Het betreft hier de omkering van subject en werkwoord.

- INV(*boolean*)

Imperatief We veronderstellen hier dat de imperatieve vorm niet alleen wordt gedefiniëerd door de modus van het werkwoord, maar ook door de afwezigheid van het subject. Merk op dat imperatief en inversie daarom niet tegelijkertijd kunnen optreden. Bovendien komt imperatief niet voor in combinatie met modale hulpwerkwoorden.

- IMP(*boolean*)

WH-informatie 'what', 'why', 'where', 'when', etc. Het betreft hier het gebruik van WH-woorden in een vraag. Er kunnen ook meerder WH-woorden voorkomen.

- WHVAL(*M_list*)

Toevoegingen tussen- of achtervoegsels zoals 'please' en tags zoals 'isn't it'.

- TAIL(*tailval*)
tailval \leadsto please | tag | nil

In Tabel 1 hebben we een overzicht gegeven van een aantal verschillende combinaties van features. Hierbij geeft STATUS de grammaticale status aan binnen het DENK-fragment. WH(+)¹¹ in combinatie met INV(+) of IMP(+) wordt uitgesloten; in alle gevallen levert dit zoiets als echo-vragen, bijv. 'Is there a diffraction pattern in which plane' of 'Switch on which button'. Dit zou misschien nog wel eens sporadisch voor kunnen komen in gesproken dialogen, in de getypte dialogen zijn we dit verschijnsel niet tegengekomen. We gaan er hieronder dus vanuit dat ze ongrammatikaal zijn. Verder zien we dat 'gewone' declaratieven worden geïnterpreteerd als RI; dit kan overigens veranderen indien een modaal hulpwerkwoord wordt gebruikt (zie ook hieronder).

We zien dat de leestekens nauwelijks invloed hebben op de functie van de uiting. Er is een afwijking, namelijk bij 4 'Adjust this lens?', maar een dergelijke uiting lijkt exceptioneel. In Tabel 2 zien we het gebruik van leestekens in getypte dialogen. Hierbij waren de proefpersonen vrij in het gebruik van leestekens. We kunnen in Tabel 2 zien dat andere leestekens dan de punt weinig worden gebruikt in vrije dialogen en dat veel uitingen niet worden afgesloten door een leesteken. Opvallend is verder het weinige gebruik van uitroepstekens, terwijl het hier dikwijls ging om opdrachten; uitroepstekens kwamen in de volgende gevallen voor:

- bij confirmaties van de opdrachtgever op de handeling van de bouwer
v.b. 'Ja!', 'Ja!!!', 'Goed!!!'
- afronding van de taak
v.b. 'Klaar!!!'

¹¹Hier betekent '+' dat de waarde van het argument in ieder geval niet nil is.

Tabel 1: De invloed van leestekens (PUNC), WH-woorden (WH), inversie van subject en werkwoord (INV), en de imperatieve modus (IMP) op de vaststelling van de illocutieve kracht van de uiting (FORCE).

	PUNC	WH	INV	IMP	Example	STATUS	FORCE
1	?	-	+	-	Is the em in HM-mode?	+	RI
2	?	-	+	+	-	-	-
3	?	-	-	-	The em is in HM-mode?	+	RI
4	?	-	-	+	Adjust this lens?	+	RI
5	?	+	+	-	Is the em in which mode?	-	-
6	?	+	+	+	-	-	-
7	?	+	-	-	Which lens is on?	+	RI
8	?	+	-	+	Adjust which lens?	-	-
9	!	-	+	-	Is the em in HM-mode!	+	RI
10	!	-	+	+	-	-	-
11	!	-	-	-	The em is in HM-mode!	+	RI
12	!	-	-	+	Adjust this lens!	+	RA
13	!	+	+	-	Is the em in which mode!	-	-
14	!	+	+	+	-	-	-
15	!	+	-	-	Which lens is on!	+	RI
16	!	+	-	+	Adjust which lens!	-	-
17	.	-	+	-	Is the em in HM-mode.	+	RI
18	.	-	+	+	-	-	-
19	.	-	-	-	The em is in HM-mode.	+	RI
20	.	-	-	+	Adjust this lens.	+	RA
21	.	+	+	-	Is the em in which mode.	-	-
22	.	+	+	+	-	-	-
23	.	+	-	-	Which lens is on.	+	RI
24	.	+	-	+	Adjust which lens.	-	-
25	.	-	+	-	Is the em in HM-mode	+	RI
26	.	-	+	+	-	-	-
27	.	-	-	-	The em is in HM-mode	+	RI
28	.	-	-	+	Adjust this lens	+	RA
29	.	+	+	-	Is the em in which mode	-	-
30	.	+	+	+	-	-	-
31	.	+	-	-	Which lens is on	+	RI
32	.	+	-	+	Adjust which lens	-	-

Tabel 2: Gebruik van leestekens in dialogen uit blokkenexperiment afhankelijk van de imperatieve modus van het werkwoord (**imp**), al dan niet inversie van werkwoord en subject (**decl** of **int**) en het gebruik van WH-woorden. (20 dialogen)

	' '	'.'	'?'	'!'	':'	';	totaal
imp	76	20	-	-	-	3	99
decl	56	101	-	3	3	6	168
int	4	1	7	-	-	-	12
WH	2	-	3	-	-	-	5
totaal	138	121	10	3	3	9	284

Tabel 3: Gebruik van leestekens in dialogen uit vluchtinformatie-experiment afhankelijk van de imperatieve modus van het werkwoord (**imp**), al dan niet inversie van werkwoord en subject (**decl** of **int**) en het gebruik van WH-woorden. (20 dialogen)

	' '	'.'	'?'	'!'	totaal
imp	-	1	-	-	1
decl	1	89	15	1	96
int	-	2	171	1	174
WH	1	4	249	-	254
totaal	2	96	435	2	535

- verbetering door de opdrachtgever van de bouwer
v.b. 'eentje maar!!', 'samen!!!', 'de andere groene.!'
- manipulatief
v.b. 'de blauwe balk moet nog een naar voor!'
- diversen
v.b. 'ik heb nog niets gekregen!'

Tabel 3 laat de resultaten zien van de vluchtinformatiedialogen waarbij, in tegenstelling tot de dialogen in Tabel 2, de proefpersonen gedwongen werden leestekens te gebruiken. Merk op dat er in Tabel 3 sprake is van slechts één imperatief; het betreft hier het geval: 'geeft u me maar van alle drie', hier een verzoek om een taalhandeling. Ook hier worden uitroepetekens weinig gebruikt:

- in een vraag van de informatiezoeker (waarschijnlijk voor extra accent)
v.b. 'Kunt u mij de aankomsttijden geven van de rechtstreekse vluchten uit Lyon frankrijk!'
- antwoord op wedervraag van informatiegever
v.b. 'Ja, dat bedoel ik!'
- dankzegging van de informatievrager
v.b. 'Bedankt!'

In praktijk lijkt de punctuatie dus weinig bij te dragen aan de interpretatie van de taalhandeling en overheersen andere features zoals het inverteren van subject en (hulp)werkwoord.

In de toekomst zullen we ook de andere combinaties bekijken, zoals de toevoeging van *please* and de *tags*. We laten dit voorlopig over aan de lezer; de uitwerking hiervan beschreven zal worden in de deliverable *Conduct*.

5.2 Het gebruik van speciale werkwoorden

Onder speciale werkwoorden verstaan we werkwoorden die niet direct betrekking hebben op domeintoestanden of -events, maar werkwoorden die dikwijls een relatie aangeven tussen de dialoogdeelnemers en de domeininformatie. Hierbij maken we onderscheid tussen enerzijds *modale hulpwerkwoorden* (*can, will, etc.*), en anderzijds wat we zullen noemen *attitude werkwoorden* (*believe, think, desire, etc.*) en *communicatieve werkwoorden* (*tell, explain, describe, ask, question, etc.*). Gemotiveerd door de informatiestroom tussen de dialoogpartners (hier het systeem S en de gebruiker U) die wordt uitgedrukt in de communicatieve werkwoorden, worden deze werkwoorden ingedeeld in twee categorieën: *tell* en *ask*. Verder worden de attitude werkwoorden ingedeeld in zogenaamde *informatie-attitudes* (*know*) en *pro-attitudes* (*want*).

In de onderstaande definitie van *VERB_CONS* (Verb-Constructie) gaan we ervan uit dat modale hulpwerkwoorden niet samen zullen voorkomen met pro-attitudes en informatie-attitudes niet samen communicatieve werkwoorden.¹² Verder zouden we strikt genomen bij de communicatieve werkwoorden moeten specificeren tussen welke partners de informatiestroom zou moeten stromen. Aangezien we er echter vanuit gaan dat er slechts twee dialoogpartners zijn, specificeren we hier alleen de actor, d.w.z. degene die de actie dient uit te voeren (zie voorbeeld (5.4d) hieronder). De lijst met markers is uiteindelijk dezelfde als die wordt gebruikt bij het feature *FORCE*.

```

• VERB_CONS(partner, verb1, partner, verb2, M_list)
verb1  ~> modal | want | nil
modal  ~> can | could | may | must | might | shall | should | will | would
partner ~> S | U | nil
verb2  ~> tell | ask | know | nil

```

Dit maakt het mogelijk constructies te behandelen zoals:

(5.4a) 'Can you tell me how to switch on the microscope?'

VERB_CONS(S, can, S, tell, marker)

(5.4b) 'May I ask you in which state the microscope is in?'

VERB_CONS(U, may, U, ask, marker)

(5.4c) 'I want to know which button controls the contrast.'

VERB_CONS(U, want, U, know, marker)

¹²Denk bijvoorbeeld aan exotische constructies als 'Can you want to increase the voltage' of 'I believe that you ask me what time it is'.

(5.4d) 'I want you to tell me how the microscope can be switched on.'
VERB_CONS(U, want, S, tell, *marker*)

(5.4e) 'Explain why there is no light at the screen!'
VERB_CONS(nil, nil, S, tell, *marker*)

(5.4f) 'Can you increase the voltage?'
VERB_CONS(S, can, nil, nil, *marker*)

(5.4g) 'Increase the voltage!'
VERB_CONS(nil, nil, nil, nil, *marker*)

Uiteindelijk dient dit feature met bijbehorende waarden omgezet te worden naar RI, RI_POS of RA. De informatie over de Verb-Constructie dient wel bewaard te worden om een passend antwoord te kunnen genereren. De genoemde categorieën spelen onder andere een belangrijke rol bij indirecte vragen en opdrachten. In de volgende subsectie zullen we aan de hand van het hulpwerkwoord *can* bekijken hoe deze werkwoorden de interpretatie van de taalhandeling kunnen beïnvloeden en welke informatie we nodig hebben om tot een coöperatieve reactie te komen.

5.3 Het modale hulpwerkwoord *can*

Aan de onderstaande voorbeelden kunnen we zien dat, onafhankelijk van de dialoogcontext, de illocutie van uitingen met het modale hulpwerkwoord *can* op verschillende manieren geïnterpreteerd kan worden.

(5.5a) Can I switch on the microscope?

(5.5b) ?Can I tell you which state the microscope is in?

(5.5c) ?Can I be told which state the microscope is in?

(5.5d) Can I ask you which state the microscope is in?

(5.5e) ?Can I know which state the microscope is in?

(5.5f) Can you switch on the microscope?

(5.5g) You can switch on the microscope.

(5.5h) ?Can you believe that the em in mP-mode?

(5.5i) Can you explain why there is no light at the screen?

(5.5j) Can the microscope switch into nP-mode?

(5.5k) Can the contrast be increased with this button?

(5.5l) Can the microscope be in nP-mode?

Hierbij zien we onder andere dat indien het subject verwijst naar de assistent, de uiting een meer imperatieve status krijgt. Dit is nog wel het meest duidelijk in de voorbeelden (5.5g) en (5.5i), maar ook voorbeeld (5.5f) doet sterk denken aan het bekende 'Can you pass me the salt, please?'. Merk overigens op dat de tags zoals 'please' hierbij extra informatie kunnen geven op illocutief niveau. We vermoeden dat de keuze voor een bepaalde interpretatie sterke regelmatigheden vertoont die onder andere samenhangen met de keuze voor het subject, en de expertise, de mogelijkheden en de machtsverhoudingen van de dialoogdeelnemers binnen een dialoogsituatie. We zullen hier niet te diep op ingaan, maar hieronder wel enkele gevallen bekijken.

In het algemeen kunnen uitingen van de vorm 'X can VP' en hun inverse 'Can X VP' geïnterpreteerd worden in termen van *possibility of permission*:

(a) *Possible(X, VP)* (, given the current state of the domain)

(b) *Permission(X, VP)*

waarbij *Possible(X, VP)* betrekking heeft op de fysieke mogelijkheid, terwijl *Permission(X, VP)* betrekking heeft op al dan niet bestaande 'verbods'-modaliteiten. Deze laatste is in het huidige prototype semantisch nog niet te behandelen en het is op dit moment ook niet duidelijk waarvoor deze toegepast zou kunnen worden in het EM-domein. Wel kunnen we deze interpretatie gebruiken om een motivatie aan te geven voor de uiteindelijke keuze die we maken voor de (indirecte) taalhandeling. In andere domeinen kan evenwel gedacht worden aan situaties waar de verbods-constraints wel direct van belang zouden kunnen zijn, bijvoorbeeld bij het waarschuwen voor irreversibele situaties (bijv. het deleten van een file of het voorkomen van een ongeluk) of het volgen van geboden of verboden (zoals verkeersregels). Merk overigens op dat, indien het subject verwijst naar de gebruiker of de assistent, de VP een domeinhandeling of taalhandeling representeert; in andere gevallen representeert de VP een domein-eventualiteit.

Voorbeeld (5.5a) kan dus geïnterpreteerd worden als:

- *Possible(U, switch on the microscope)?* of

- *Permission(U, switch on the microscope)?*

De eerste interpretatie is niet voor de hand liggend; immers, aangezien iedere dialoogdeelnemer in veel gevallen expert is over zijn eigen mogelijkheden, weet de gebruiker dat de assistent niet of moeilijk kan beoordelen of de gebruiker in staat is om de handeling uit te voeren. De tweede interpretatie lijkt nu meer plausibel, omdat er mogelijk redenen zijn dat de microscoop nog niet aangezet zou mogen worden. Een probleem is echter dat we in beide gevallen geen correcte interpretatie kunnen doen omdat het systeem geen kennis heeft over de capaciteiten van de gebruiker en, zoals we al eerder aangaven, geen verbods-modaliteiten kent. We kunnen de uiting echter wel indirect interpreteren, waarbij dan bijvoorbeeld de mogelijkheid bestaat om de gemaakte interpretatie terug te melden.

In het eerste geval lijkt een plausibele indirecte interpretatie er een te zijn waarbij de modaliteit naar binnen schuift en dus de volgende zwakkere versie oplevert:

- *Possible (switch on the microscope)?*

oftewel een vraag naar mogelijkheid (RI_POS) met inhoud de actie 'switch on the microscope'.

In het tweede geval zou men vanuit Griceaans oogpunt kunnen redeneren dat wanneer iemand ergens toestemming voor vraagt, hij of zij dan ook de bijbehorende actie uitgevoerd wil hebben, wat ertoe zou kunnen leiden dat de vraag indirect wordt geïnterpreteerd als RA. Maar het zal duidelijk zijn dat het uitvoeren van de handeling meer consequenties heeft dan het beantwoorden van de vraag. Bovendien biedt het beantwoorden van de vraag het systeem nog altijd de mogelijkheid te vragen of de actie ook daadwerkelijk uitgevoerd moet worden.

In de keuze voor het antwoord op de oorspronkelijke vraag hebben we in het positieve geval bijvoorbeeld de volgende mogelijkheden:

(5.6) A: Can I switch on the microscope?

B1: Yes

B2: Yes, it is possible to switch on the microscope.

B3: Yes, the microscope can be switched on.

B4: Yes, it is possible that you switch on the microscope.

B5: Yes, you can switch on the microscope.

B6: Yes, do you want me to switch it on?

Respons B5 geeft weinig extra informatie ten opzichte van het eenvoudigere Yes. De tweede respons (B2) heeft als voordeel dat de assistent terugmeldt welke indirecte interpretatie is gekozen. Indien de gebruiker dit niet zo bedoeld heeft kan de oorspronkelijke uiting nog altijd nog anders worden geformuleerd.¹³ In het negatieve geval kunnen soortgelijke overwegingen gelden.

Wanneer we het subject wijzigen van de gebruiker in de assistent zoals in voorbeeld (5.5f) of voorbeeld (5.5g) ('Can you VP?' of 'You can VP.'), lijkt de voorkeur voor de (indirecte) interpretatie uit te gaan naar een request for action, in het bijzonder bij de declaratieve vorm. In tegenstelling tot het vorige geval lijkt de declaratieve vorm vooral de betekenis van het opheffen van het verbod op een handeling te hebben, wat de RA interpretatie zou kunnen verklaren. Merk verder op dat de vragende vorm niet de associatie oproept met het vragen om toestemming. Dit zou natuurlijk ook vreemd zijn, omdat de spreker dan toestemming zou vragen voor de assistent aan de assistent zelf om de handeling uit te voeren. We bekijken hier nog een paar mogelijke reacties op de interrogatieve vorm:

(5.7) A: Can you VP?

B1: ?Yes

B2: Yes, done.

B3: No (, because p.)

B4: Yes, do you want me to VP?

¹³Dit kan leiden tot interessante uitbreidingen van de mogelijkheden, zoals het toevoegen van correcties No, I mean, may I switch on the microscope, maar dit valt buiten het ontwerp van het huidige prototype.

Tabel 4: Interpretatie van 'X can VP' (d = declaratief) or 'Can X VP' (i = interrogatief). Zie verder de bijbehorende tekst.

X	VP					
	domain		communicatief		info-attitude	
	d	i	d	i	d	i
U	RI_POS	RI_POS	RI	RI	*	*
S	RA	RI_POS	RI	RI	*	*
domain object	RA	RI_POS	*	*	*	*

Om de ambiguïteit na de respons te vermijden in de interpretatie (RI of RA) is in B2 extra informatie toegevoegd ('done'). Dit betekent dat we in de talige reactie in ieder geval een onderscheid moeten kunnen maken tussen een directe reactie op een imperatief (domeinhandeling) en de indirecte reactie op een 'Can you VP?' uiting. Welke weg uiteindelijk gevolgd moet worden om tot een adequate reactie te leiden is hier echter niet aan de orde en zal verder bekeken worden binnen Conduct. We hebben hier alleen willen aangeven welke overwegingen hierbij een rol kunnen spelen.

In het geval waarbij het werkwoord in de VP vervangen is door een zogenaamd communicatief werkwoord (tell) kan de uiting pragmatisch alleen zinvol geïnterpreteerd worden als het subject de assistent betreft. Dikwijls bevat de uiting dan een WH-feature¹⁴ wat dus ook vertegenwoordigd is in het geannoteerde segment. Met andere woorden, deze uitingen kunnen behandeld worden als RI's waarbij het feature WH de informatie-gap aangeeft, inclusief 'whether' voor ja/nee-vragen.

(5.8) A: Can you tell me whether p?

B1: ?No (or ?Yes)

B2: No, I cannot

B3: (No,) not p

B4: Indeed, p

Ook hier kan de keuze voor de directe en indirecte interpretatie duidelijk gemaakt worden door het toevoegen van extra informatie.

Indien het subject X een domeinobject is, kan de VP ook een domeintoestand of (een passieve vorm van) een actie zijn. Ook hier lijkt de declaratieve vorm meer in de richting te gaan van een actie die door de assistent uitgevoerd dient te worden, terwijl de interrogatieve vorm meer een vraag om informatie lijkt uit te drukken. In de declaratieve gevallen is de keuze echter niet eenduidig te maken, maar gevoelsmatig leiden deze gevallen uiteindelijk tot een RA en de interrogatieve versie tot een RI_POS. We illustreren dit aan de hand van de volgende voorbeelden:

(5.9a) The microscope can switch into nP-mode.

(5.9b) Can the microscope switch into nP-mode?

(5.9c) The contrast can be increased with this button.

¹⁴Dit is echter niet altijd het geval, bijv. 'Can you give me the reason for the shift of the electron beam?'

(5.9d) Can the contrast be increased with this button?

Het zou hier te ver voeren op alle details in te gaan en het zal duidelijk zijn dat de discussie hierover nog niet is afgelopen. We hebben de voorlopige resultaten samengevat in Tabel 4. In Tabel 4 worden overigens niet alle gevallen afgedekt die strikt genomen wel mogelijk zijn volgens de VERB_CONS definitie¹⁵; we gaan hier alleen uit van pragmatisch acceptabele gevallen.

6 Prominentie en WH-vragen

Het laatste onderwerp dat we hier behandelen is het gebruik van prominentie in de uitingen van de gebruiker. We merken hierbij vooraf op dat prominentie niet zal leiden tot extra pragmatische features. Maar aangezien het gebruik van prominentie belangrijke gevolgen heeft voor het genereren van extra informatie in responsies op vragen en daarmee dus een duidelijk pragmatisch aspect is van de uiting, hebben we dit hier toch opgenomen.

Reeds in Bäuerle (1979) wordt geobserveerd dat door middel van intonatie zogeheten 'achterliggende' WH-vragen kunnen worden geassocieerd met ja/nee-vragen. Neem bijvoorbeeld de volgende ja/nee-vraag inclusief intonatiepatroon (waarbij het subscript *P* aangeeft dat het betreffende zinsdeel intonatieel prominent is):

(6.1) 'Did John_P open the door?'

Er lijkt nu een verschil te zijn in acceptabiliteit tussen de volgende twee responsies op deze vraag:

(6.2a) 'No, Mary opened the door.'

(6.2b) 'No, John closed the door.'

Respons (6.2b) lijkt minder acceptabel dan respons (6.2a).¹⁶ Dit kan verklaard worden uit het feit dat de ja/nee-vraag is geassocieerd met de volgende achterliggende WH-vraag: 'Who opened the door?' Merk op dat respons (6.2a) een antwoord geeft op de achterliggende WH-vraag, terwijl (6.2b) dat niet doet.

Voor het DENK-systeem zouden achterliggende WH-vragen relevant kunnen zijn omdat ze ook geïnduceerd kunnen worden door bepaalde syntactische constructies. Een voorbeeld hiervan is de *it*-cleft:

(6.3) 'Was it John who opened the door?'

Wederom lijkt (6.2a) een acceptabelere respons op deze uiting dan (6.2b). Het is echter zeker niet zo dat alle syntactische constructies die een zinsdeel naar het begin van de zin verplaatsen deze functie hebben. Neem bijvoorbeeld:

(6.4) 'John, did he open the door?'

¹⁵Dit zijn bijvoorbeeld pragmatisch vreemde gevallen als 'Can you tell you ...?' of 'Can you ask me ...?'.
¹⁶Merk op dat het omgekeerde geldt wanneer we uitgaan van het volgende intonatiepatroon: 'Did John open_P the door?'

De functie van links-dislocatie is om een referent de topic te maken (zie Rats, 1996). Topichood van John lijkt de onderliggende WH-vraag 'Who opened the door' uit te sluiten: het feit dat John het topic is, betekent dat de zin gaat over John (zie bijv. Reinhart, 1981), d.w.z. er wordt informatie ingewonnen over John, en in het bijzonder betreffende de vraag of John de deur heeft geopend. Als het antwoord hierop negatief is, dan ligt het niet voor de hand om vervolgens te gaan spreken over andere personen die de deur wel geopend hebben (zoals in 'No, Mary opened the door', waarmee een antwoord wordt gegeven op de WH-vraag 'Who opened the door?'), aangezien de dialoogparticipanten het over John hadden.

Laten we nu kijken hoe we de notie van een onderliggende WH-vraag kunnen formaliseren in geannoteerde Typen Theorie. Hierbij maken we gebruik van het idee dat aan een uiting twee betekenissen kunnen worden toegekend: enerzijds de gewone betekenis en anderszijds de betekenis van een eventuele onderliggende vraag.¹⁷ Beide betekenissen zullen worden gerepresenteerd met een zogeheten geannoteerd segment. De betekenissen worden gekoppeld aan één uiting doordat we beschikken over een uitingsreferent waaraan maximaal twee geannoteerde segmenten kunnen worden gekoppeld. Het geannoteerde segment dat correspondeert met de gewone semantisch betekenis kan met behulp van de volgende service worden opgevraagd (zie Ahn et al., 1996):

(6.5) get_as(+U:UtteranceReferent, -AS:Annotated_Segment)

Gegeven een uitingsreferent, retourneert deze service het geannoteerde segment dat de gewone betekenis van de uiting representeert. Voor het opvragen van het andere geannoteerde segment stellen we de volgende (nieuwe) service voor:

(6.6) get_as_sub(+U:UtteranceReferent, -AS:Annotated_Segment)

Hierbij geeft sub aan dat het gaat om de 'onderliggende (WH-vraag)' betekenis van de uiting.

Voor het representeren van de betekenis van vragen in geannoteerde segmenten maken we gebruik van een idee dat is geopperd in Cohen (1929). Cohen merkt op dat WH-vragen corresponderen met propositionele functies, d.w.z. proposities met een of meer ongebonden variabelen. Vergelijk bijvoorbeeld 'Wat is de som van 12 en 13?' met $x = 12 + 13$.

In Typen Theorie corresponderen de voornoemde ongebonden variabelen met gaps (markers).¹⁸ De representatie van een onderliggende WH-vraag kan dus worden verkregen door de stukjes in een segment die corresponderen met de prominente delen van de uiting te vervangen door gaps (die gerepresenteerd worden door hoofdletters). Neem bijvoorbeeld de vraag 'Who walks?'. Bij deze vraag hoort het volgende geannoteerde segment (zie ook voorbeeld (5.2)):

(6.7) <[X:person, P:walk-X], [FORCE(RI, [X])]>

¹⁷We gebruiken hier betekenis in de brede zin: deze omvat zowel semantiek als de functie van de uiting. In Rooth (1985) wordt ook twee typen van betekenis aan zinnen (terwijl wij spreken van uitingen) toegekend, maar hier is sprake van betekenis in enge zin: de 'ordinary semantic value' en de 'focus semantic value'.

¹⁸Zie Ahn (1994) en Piwek (1996).

Laten we nu kort schetsen welke rol onderliggende vragen spelen in het generatieproces. We gaan er vanuit dat naar aanleiding van de ULF-representatie van de uiting een uitingsreferent wordt geïntroduceerd waaraan twee geannoteerde segmenten zijn gekoppeld: (a) het geannoteerde segment (afgekort AS) voor de gewone betekenis en (b) het geannoteerde segment voor de onderliggende vraag.¹⁹

Om een reactie te genereren wordt allereerst op het AS voor de gewone betekenis Resolve toegepast. Vervolgens wordt een respons op deze AS berekend. Stel nu dat de AS een ja/nee-vraag betreft en dat het antwoord op deze ja/nee-vraag negatief is. In dat geval, zal het systeem bekijken of het eventueel een antwoord kan geven op de onderliggende WH-vraag die gerepresenteerd wordt door het tweede AS. Dit betekent kortweg dat op het tweede AS Resolve wordt toegepast om de referentiële uitdrukkingen erin te resolveren, waarna vervolgens een antwoord wordt berekend op de onderliggende vraag. Dit antwoord wordt dan samen met het negatieve antwoord op de ja/nee-vraag gecommuniceerd naar de gebruiker.

Merk op dat we prominentie die onderliggende WH-vragen induceert niet hebben gedefinieerd als betrekking hebbende op specifieke CTT-variabelen (in DRT termen, discourse referenten) die in het segment voorkomen. Dit zou ons namelijk in problemen brengen bij de representatie van 'John likes himself_P'. Hier wordt er één variabele geïntroduceerd voor John. Als we zouden aangeven dat deze referent prominent is, dan zouden we geen onderscheid kunnen maken tussen *John* en *himself*, en dat is wel nodig om aan te geven dat de prominentie juist net betrekking heeft op de laatst genoemde expressie.

We hebben er tevens van afgezien om een uiting waarin prominentie voorkomt te analyseren als bestaande uit een segment waarin de prominente delen zijn vervangen door gaps en een annotatie die aangeeft wat de invulling van deze gaps is (hieronder aangegeven door het feature BIND; vergelijk Pierrehumbert & Hirschberg, 1990:289). In dat geval zou bijvoorbeeld 'John_P walks' corresponderen met:

(6.8) <[X: person, P:walks-X], [BIND(X, john)]>

We hebben om de volgende twee redenen afgezien van de hierboven geschetste benadering:

1. Het is niet duidelijk hoe quantificationale expressies die prominent zijn gerepresenteerd zouden moeten worden (bijvoorbeeld '[Every woman]_P walks').
2. Er treedt inferentie op tussen de representatie van prominentie en de operaties die Resolve uitvoert op geannoteerde segmenten. Neem bijvoorbeeld 'The tall_P man walks'. De onderliggende vraag correspondeert in dit geval met de volgende propositionele functie: The Y man walks. We kunnen dit representeren in geannoteerde Typen Theorie, maar dan alleen met de representatie die voorhanden is voor Resolve is toegepast. De representatie ziet er als volgt uit:

(6.9) <[X:man, M:man-->*_P, Q:M-X, P:walk-X], [RB(X), BIND(M, tall)]>

¹⁹(b) is natuurlijk niet bij alle uitingen voorhanden, en in het bijzonder niet bij die uitingen waarbij geen prominentie is gemarkeerd.

Na toepassing van Resolve wordt *X* geïnstantieerd met een concrete CTT-variabele, waardoor het geannoteerde segment niet meer correspondeert met vraag die wordt uitgedrukt door *The Y man walks* (oftewel: *Wat voor soort man loopt er?*), omdat Resolve voor *X* een specifieke variabele heeft geïntroduceerd die correspondeert met *The tall man*.

Door een apart AS te introduceren voor de onderliggende vraag kan dit probleem vermeden worden: Resolve kan nu immers afzonderlijk worden toegepast op het AS voor de gewone betekenis los van het AS voor de onderliggende vraag.

Een begrip dat vaak in een adem met de notie prominentie wordt genoemd is het zinstopic. Zoals reeds eerder aangegeven, zullen we er van uitgaan dat het zinstopic dat object is waarover de zin gaat. Welk object de rol van zinstopic speelt is o.a. van invloed op anaforenresolutie (zie Ariel, 1990). Momenteel zien we geen directe toepassing van de notie zinstopic in het DENK-systeem. Indien nodig, zou een zinstopic gerepresenteerd kunnen worden door markering van de corresponderende CTT-variabele in het geannoteerde segment.

7 Overige features

Alhoewel we te maken hebben met slechts twee dialogedeelnemers is het soms noodzakelijk te achterhalen wie verantwoordelijk is voor het introduceren van een bepaalde marker. Dit kan enerzijds gebeuren door het uiten van bepaalde informatie, maar anderzijds ook door het uitvoeren van een actie. Dit speelt vooral een rol zolang de informatie nog niet volledig verwerkt is en zich dus nog op de Pending Context bevindt. We geven daarom bij iedere marker aan door wie deze is geïntroduceerd, het systeem *S* of de gebruiker *U*.²⁰ Hiervoor maken we gebruik van het feature RP (Responsible Participant) dat bestaat uit twee argumenten, a. de verantwoordelijke participant en b. de bijbehorende lijst van markers.

• RP(*partner*, *M_list*)
partner \rightsquigarrow S | U

Bijvoorbeeld:

(7.1) User: 'Switch off the lens!'
 <[P:switchoff-lens318], [FORCE(RA, [P]), RP(U, [x318, P])]>

Merk op dat *x318* een concrete lens is die ingevuld is door Resolve en dat deze invulling tevens heeft plaatsgevonden in het RP-feature. Indien deze actie succesvol wordt uitgevoerd door het systeem (en daarmee dus een bewijs heeft (hier *p32*) dat *lens318* is uitgezet) wordt deze informatie toegevoegd aan het geannoteerde segment:

(7.2) <[p32:switchoff-lens318],
 [FORCE(RA, [P]), RP(U, [x318]), RP(S, [p32])]>

²⁰Momenteel wordt tevens gewerkt aan een alternatieve representatie waarbij de informatie wie een bepaalde uiting heeft gedaan direct wordt gekoppeld aan de uitingsreferent. In dat geval hoeft deze informatie dus niet in het geannoteerde segment te worden opgeslagen.

Nadat het succes van de actie is teruggekoppeld naar de gebruiker, bijv. door de uiting 'Done', kan de informatie doorgeschoven worden naar de dialoogcontext (als deel van de Common Ground).

8 Conclusies

We hebben in dit rapport een overzicht gegeven van een verzameling pragmatische features die van belang zijn voor het bepalen van de betekenis van de uiting, zowel wat betreft de waarheidsconditionele betekenis, als de functionele betekenis in termen de taalhandeling en de te verwachten reactie van het systeem. Alhoewel we getracht hebben zo volledig mogelijk te zijn in onze opsomming, verwachten we geenszins dat we de features uitputtend behandeld hebben. Een belangrijk deel van datgene wat we hier gepresenteerd hebben is nog volop in onderzoek en er zijn daarom in de toekomst zeker nog veranderingen en uitbreidingen te verwachten. We hopen echter hiermee wel aangegeven te hebben dat (en hoe) pragmatische features een essentiële rol spelen in het verwerkingsproces van en reactieproces op een uiting.

Een deel van de features die we gepresenteerd hebben is gebaseerd op empirische data, maar een ander deel is voornamelijk gebaseerd op onze eigen intuïties en we zouden hierin uiteraard meer evenwicht in willen brengen. Het bekijken van empirische dialoogdata tussen twee menselijke participanten heeft echter belangrijke problemen, vooral daar waar het gaat om het oplossen van waarheidsconditionele ambiguïteiten. Ten eerste is het moeilijk zo niet onmogelijk om als onderzoeker van de empirische data bewust te zijn van alle subtiele betekenisverschillen die zich in een uiting of woord kunnen voordoen tijdens een interactie. Ten tweede speelt context een bijzonder essentiële, maar moeilijk te doorgronden rol in het desambigueringsproces en het lijkt vrijwel ondoenlijk deze aspecten bij te houden bij het bestuderen van de data. Het lijkt vrijwel onvermijdelijk hierbij gebruik te moeten maken van bijvoorbeeld een begrip als 'topic', maar hoe bijvoorbeeld eerder ingevoerde concepten in de dialoog, de taak van de participanten, de discourse structuur, de achtergrondkennis en de omgeving hierbij een rol spelen is een vraag waar, voor zover ons bekend, nog geen concrete invulling aan is gegeven.

9 Referenties

- AHN, R. (1994) Communicating Contexts: A Pragmatic Approach to Information Exchange, In: P. Dybjer, B. Nordström & J. Smith (eds), *The proceedings of the BRA workshop: Types of Proofs and Programs*, Springer Lecture Notes in Computer Science vol. 996. Springer Verlag, Berlin.
- AHN, R. & BORGHUIS, T. (1996) Chronicle. DENK-report 96/23, SOBU, Tilburg.
- AHN, R., KIEVIT, L. & PIWEK, P. (1996) Sherlock. DENK-report 96/19, SOBU, Tilburg.
- AHN, R., KIEVIT, L., RENTIER, G. & VERLINDEN, M. (1995) Sherlock DENK-report 95/12, SOBU, Tilburg.
- ARIEL, M. (1990) Accessing Noun-Phrase Antecedents, Routledge: London and New York
- BÄUERLE, R. (1979) Questions and answers. In: R. Bäuerle, U. Egli & A. von Stechow, eds., *Semantics from different points of view*, Springer-Verlag, Berlin: 61-74
- BEUN, R.J. (1987) Transcripties terminal-dialogen 1987. Deel I, II en III. IPO-rapport 608, IPO, Eindhoven.
- BEUN, R.J. & KIEVIT, L. (1995) Resolving definite expressions in DENK. DENK-report 95/16, SOBU, Tilburg.
- BUNT, H.C. (1995) Semantics and Pragmatics in the Δ ELTA system. In: L. Dybkjaer (ed.) *Topics. Proc. of the Second Dialogue and Discourse Workshop*, Dublin, April 1995. CCI, Roskilde University, ISBN 87-7349-288-4, 1-27.
- COHEN, F.S. (1929) What is a question? *The Monist* 39. 350-364
- CREMERS, A. (1993) Transcripties terminal-dialogen blokken-experiment, 1993. DENK-report 95/11, SOBU, Tilburg.
- KIEVIT, L. (1996) ULF. DENK-report 96/27, SOBU, Tilburg
- KIEVIT, L. (1997) te verschijnen.
- PIERREHUMBERT, J. & J. HIRSCHBERG (1990) The Meaning of Intonational Contours in the Interpretation of Discourse. in P.R. Cohen, J. Morgan & M.E. Pollack (eds), *Intentions in Communication*, Cambridge, MA; MIT Press. 271-311
- PIWEK, P. (1995) Annotated Type Theory for Information Representation. DENK-report 95/17, SOBU, Tilburg.
- PIWEK, P. (1996). Questions, Answers and Context in Constructive Type Theory, IPO Report no. 1125, Eindhoven (Presented at CLIN VII).

- PIWEK, P. (1997) Context Change in Dialogue Games. (te verschijnen).
- RATS, M. (1996) Topic Management in Information Dialogues, ITK Dissertation Series 1996-1, Tilburg University.
- REINHART, T. (1981) Pragmatics and linguistics: an analysis of sentence topics. *Philosophica*, 27. 53-94
- ROOTH, M. (1985) *Association with Focus*. Ph.D. diss., UMass, Amherst.
- VERBEEM, M. (1996) An evaluation of Resolve. DENK-report 96/28, SOBU, Tilburg.
- VERLINDEN, M. (1996) Development of the core syntax for DENK. DENK-report 96/22, SOBU, Tilburg.
- VERLINDEN, M. (1996) Semantische en pragmatische elementen in ULF. DENK-report 96/26, SOBU, Tilburg.

10 Appendix: Overzicht pragmatische features

Requires Binding (p.7)

• **RB**(*list_of_M_list*)

list_of_M_list \rightsquigarrow [*M_list* | *list_of_M_list*] | []

M_list \rightsquigarrow [*marker* | *M_list*] | []

Marker Type (p.7)

• **MT**(*def_list*)

def_list \rightsquigarrow [(*marker, def_type*) | *def_list*] | []

def_type \rightsquigarrow *def_art* | *subst_demon* | *attr_demon* | *nom_pronoun* | *one_indef*
| *one_def* | *one_demon*

Unresolved Object (p.7)

• **UO**(*marker, O_list*)

O_list \rightsquigarrow [*object* | *O_list*] | []

Type Restriction (p.8)

• **TR**(*list_of_TypRes*)

list_of_TypRes \rightsquigarrow [(*marker.TypeRestriction*) | *list_of_TypRes*] | []

TypeRestriction \rightsquigarrow *disj*(*list_of_indexen, list_of_Vtype*) | *leq*(*Vtype*) | *geq*(*Vtype*)

list_of_indexen \rightsquigarrow [*index* | *list_of_indexen*] | []

list_of_Vtype \rightsquigarrow [*Vtype* | *list_of_Vtype*] | []

Vtype \rightsquigarrow *marker* | *type*

Temporele Informatie (p.10)

• **TEMP_TAGS**(*connective, temp_mark, temp_mark*)

connective \rightsquigarrow *while* | *before* | *until* | *after* | *when* | *since* | *nil*

temp_mark \rightsquigarrow *past_perfect*(*marker, marker*) | *past_imperfect*(*marker*)
| *present_perfect*(*marker*) | *present_imperfect*(*marker*) | *nil*

Illocutieve kracht (p.11)

• **FORCE**(*f_name, M_list*)

f_name \rightsquigarrow *RI* | *RI_POS* | *RA*

Punctuatie (p.12)

• **PUNC**(*clospunctual*)

clospunctual \rightsquigarrow *period* | *questmark* | *exclmrk* | *nil*

Subject-Verb Inversie (p.12)

• **INV**(*boolean*)

Imperative Mood (p.12)

• **IMP**(*boolean*)

WH-vragen (p.12)

• **WHVAL**(*M_list*)

Tussen- en achtervoegsels (p.12)

• **TAIL**(*tailval*)

tailval \rightsquigarrow please | tag | nil

Verb-Constructies (p. 15)

• **VERB_CONS**(*partner*, *verb1*, *partner*, *verb2*, *M_list*)

verb1 \rightsquigarrow modal | want | nil

modal \rightsquigarrow can | could | may | must | might | shall | should | will | would

partner \rightsquigarrow S | U | nil

verb2 \rightsquigarrow tell | ask | know | nil

Responsible Participant (p.23)

• **RP**(*partner*, *M_list*)

partner \rightsquigarrow S | U