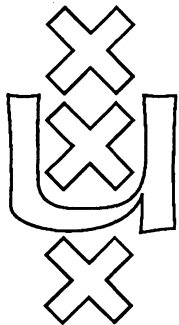


**Institute for Language, Logic and Information**

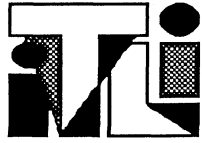
**EEN RELATIONELE SEMANTIEK VOOR  
CONCEPTUEEL MODELLEREN –  
HET RL-PROJECT**

Peter van Emde Boas

ITLI Prepublication Series  
X-89-06



**University of Amsterdam**



**Instituut voor Taal, Logica en Informatie**  
**Institute for Language, Logic and Information**

Faculteit der Wiskunde en Informatica  
(Department of Mathematics and Computer Science)  
Plantage Muidergracht 24  
1018TV Amsterdam

Faculteit der Wijsbegeerte  
(Department of Philosophy)  
Nieuwe Doelenstraat 15  
1012CP Amsterdam

**EEN RELATIONELE SEMANTIEK VOOR  
CONCEPTUEEL MODELLEREN –  
HET RL-PROJECT**

Peter van Emde Boas  
Department of Mathematics and Computer Science  
University of Amsterdam

# Een relationele semantiek voor conceptueel modelleren - het RL project

Peter van Emde Boas\*

*Faculteit Wiskunde en Informatica, Universiteit van Amsterdam,  
Plantage Muidergracht 24, 1018 TV Amsterdam  
Opgedragen aan Frans Remmen ter gelegenheid van zijn emeritaat aan de TUE  
6 December 1989*

## Abstract

Het RL project dat wordt uitgevoerd aan de Faculteit Wiskunde en Informatica van de Universiteit van Amsterdam streeft naar een geïntegreerde weergave van verschillende vormen van informatie in één taal. Gegeven een beschrijving van de wereld in haar relevante aspecten en een vraag over deze wereld dient het systeem deze vraag te vertalen in een query die op een klassieke relationele gegevensbank kan worden losgelaten, en vervolgens de volledige verzameling van antwoorden oplevert. Uitgangspunt bij de ontwikkeling is geweest dat de taal een wiskundige semantiek dient te bezitten die het predicaat *verzamelings theoretisch model* terecht mag dragen. Dit artikel behandelt achtereenvolgens de semantische visie van waaruit het project is ontstaan, de voorgeschiedenis van het project en de huidige gang van zaken binnen het project.

## 1 Het waarom van de mathematische semantiek

Laat ik beginnen met een persoonlijke bekentenis. De auteur van dit artikel is een vreemdeling in de wereld van het conceptueel modelleren. Als iemand mij zou vragen dit begrip te definiëren zou ik daar moeite mee hebben. Een vervolgens uitgesproken definitie zou hoogstwaarschijnlijk zondigen tegen veel van de in de wereld van de conceptuele modelleerders gangbare uitgangspunten. Het is echter zo gesteld dat een project dat binnen mijn onderzoeksgroep in uitvoering is tot dit gebied behoort. Aan de hand van een bespreking van dit project hoop ik duidelijk te maken hoe ik, als wiskundig opgeleide theoretisch informaticus tegen de problematiek van het conceptueel modelleren aankijk.

Bij het conceptueel modelleren gaat het om het vervaardigen van iets dat ten grondslag kan liggen aan of behulpzaam kan zijn bij het ontwikkelen van een informatie systeem. Een informatiesysteem is in mijn ogen een gemechaniseerd systeem dat in principe de vragen van een gebruiker voorziet van de zelfde antwoorden als de menselijke beheerder van een papieren informatiesysteem gegeven zou hebben, en dan bij voorkeur in minder tijd en tegen aanmerkelijk geringere kosten. In hoeverre de systemen die als informatiesystemen in de wereld bekend zijn zich aan de pragmatische aspecten van deze omschrijving houden

---

\*tevens CWI-AP6, Amsterdam

is een onderwerp waar ik het vandaag niet over wil hebben. Het gaat mij vandaag om de rol van het conceptuele model.

Bij een klassieke papieren informatiesysteem wordt aan het conceptuele model geen belangrijke rol toegedacht. De administratie beschrijft de wereld in haar (voor de bezitter van de administratie) relevante aspecten, en de menselijke gebruikers weten kennelijk daarmee overweg te gaan. Zij interpreteren de op kaarten geschreven teksten relatief de wereld zoals zij die dagelijks observeren, en kunnen, mochten zich omstandigheden voordoen waarbij deze interpretatie problematisch wordt, een voor de hand liggende uitweg vinden. Hierbij worden zij zonedig ondersteund door informatie op de kaarten die zich niet houdt aan het standaard formaat. Tenslotte is de administratie zelf een bestaand object, dat mogelijk als haar eigen model kan optreden.

Het gebruik van de computer heeft ons wereldbeeld in dit opzicht aanmerkelijk vertoebeld. Opgeslagen informatie bestaat alleen nog maar uit bits en tekens. De opslagstructuur ervan werkt als een soort ontcijfersleutel, die een vertaling naar de electronische versie van de archiefkaart mogelijk maakt. Deze vertaling is slechts dan mogelijk als ieder element zich houdt aan de daartoe strekkende formaatregels. Het extra krasje op een kaart dat een incidentele afwijking moet verduidelijken bestaat niet meer (tenzij in de ontwerpfase reeds was voorzien dat dit krasje op die plaats had kunnen optreden en dan is het geen uitzondering meer).

Het optreden van stringente regels voor het formaat van opgeslagen informatie, dwingt ons er over na te denken wat we willen opslaan en waarom we dat willen doen. Deze vragen leiden tot de behoefte aan een model. Onze administratie is geschreven in een taal die iets moet zeggen over de wereld; dit is echter niet de echte wereld maar slechts een afspiegeling daarvan. De wereld zelf is te groot, te onregelmatig en te ongrijpbaar. Het model bevat de relevante structuren, het is bevatbaar en bovenal, het houdt zich aan de beperkingen die wij eraan menen op te moeten leggen. De echte wereld lijkt zich daaraan nooit te willen houden.

In deze visie bevind ik mij nu wel degelijk op bekend terrein: de relatie tussen een taal en een wiskundig model waarin de uitdrukkingen in de taal worden geïnterpreteerd. Dit is het terrein van de modeltheoretische semantiek. Voor mij is een conceptueel model niets anders dan een semantisch model voor een taal. De taal zal bij deze toepassing in het algemeen afkomstig zijn uit de wereld van de informatiesysteemontwikkelaars. Het model is een wiskundige structuur waarover wij dingen zouden kunnen bewijzen. Als wiskundige laat ik de relatie met de echte wereld verder buiten beschouwing.

Ik verwacht niet dat ik met deze positiekeuze ver af sta van wat in de visie van de deskundigen op het gebied van informatiesystemen gebruikelijk is. De discrepantie treedt echter weer in het daglicht als we ons gaan bezinnen op de eisen die gesteld moeten worden aan de structuur van de taal, het model en de semantiek die beide verbindt. Het is op dit punt dat de wiskundige gewend is veel strengere eisen te stellen dan de meer praktisch georiënteerde systeemontwerper. Het gaat hier over eisen waarvan de practicus de relevantie wellicht niet zal doorzien zolang hij niet geconfronteerd wordt met de mogelijke gevolgen van het zich er niet aan houden.

De eisen die mij voor ogen staan laten zich als volgt samenvatten:

- De taal moet goed gedefinieerd zijn, bij voorkeur op basis van een simpel te begrijpen formalisme zoals ontwikkeld in de leer van de formele talen en de automatentheorie [9]

- Het model moet in principe te beschrijven zijn als een verzamelingstheoretische structuur, dan wel een structuur binnen een wiskundige theorie waarvan de wiskunde de semantiek reeds door en door begrepen heeft
- De relatie die de toelaatbare uitdrukkingen in de taal verbindt met hun interpretatie in het model dient zich te houden aan het *compositionaliteitsbeginsel* : *De betekenis van een samengestelde uitdrukking is samengesteld uit de betekenissen van haar delen*

Hoewel de eerste twee eisen reeds de nodige beperkingen opleggen is het vooral de compositionaliteitseis die het zware werk doet. Bewezen kan worden dat zij in feite de eerdere twee eisen impliceert. Voor de analyse waaruit dit volgt verwijs ik naar de monografie van Theo Janssen [5] [6], en voor een meer ideologische beschouwing naar [7]. De eerste eis voegt echter een extra pragmatisch accent toe: de taal mag niet te ingewikkeld worden, want zij moet uiteindelijk op een praktische computer verwerkt kunnen worden. Met de tweede eis wil ik mij afzetten tegen een verschijnsel dat ik met toenemende frequentie observeer in de wereld van de conceptuele modellen en de artificiele intelligentie: het optreden van *talige* modellen.

Wat ik met een talig model bedoel kan ik het beste illustreren aan een klassiek voorbeeld. Een van de oudste formalismen voor het begrip effectieve berekenbaarheid is gebaseerd op de Lambda calculus als ontwikkeld door Church. Uitdrukkingen in deze calculus beogen abstracte functies en toepassingen daarvan op argumenten te beschrijven. Het vervelende is echter dat in deze calculus geen verschil bestaat tussen functies en argumenten, hetgeen de perverse mogelijkheid opent om functies los te laten op zichzelf. De wiskundigen waren zich destijds terdege bewust dat deze optie aanleiding geeft tot tegenstrijdigheden. Kortom: men beschikte over een fraaie calculus maar niet over een consistent wiskundig model waarover deze calculus zou moeten handelen.

Nu bezitten de wiskundigen een techniek om voor een calculus een model te vervaardigen als ze nog niet over zo'n model beschikken. Men neme de formules in het formele systeem zelf, en men verklare twee formules equivalent indien zij volgens de regels van de calculus gelijk moeten zijn. Het model bestaat nu uit de equivalentieklassen van formules onder deze equivalentie relatie. Als de calculus redelijk in elkaar zit en consistent is levert deze termmodel constructie een model op dat aan alle eisen van redelijkheid voldoet. Alle operaties uit de calculus kunnen in het model worden geïnterpreteerd, en alle gelijkheden die de calculus vereist zijn per definitie waar in het model. Zelfs als de calculus inconsistent mocht wezen werkt de termmodel constructie - zij levert in dit geval een model op dat slechts uit één element bestaat.

Het bezwaar van een termmodel is echter dat je door naar het model te kijken niets kan leren wat je in de calculus niet ook reeds geleerd kon hebben. Het termmodel helpt niet om nieuwe inzichten te krijgen in de structuren waarover de calculus handelt. En om terug te keren tot mijn voorbeeld: het heeft tot de late zestiger jaren geduurd alvorens Dana Scott erin is geslaagd echte wiskundige modellen voor de Lambda calculus te construeren; ons begrip voor deze calculus is daardoor aanmerkelijk toegenomen. Voor verdere informatie verwijs ik naar de voortreffelijke monografie van Henk Barendregt [2].

Wat mij nu tegenstaat aan veel zich semantisch noemend werk in de wereld van de conceptuele modellering is het gebruik van semantische structuren die bij nadere beschouwing van zuiver syntactische aard blijken te zijn. Een dergelijke semantiek vertelt niets meer dan de formules toch al vertellen en maakt op deze wijze zich zelf overbodig. Dat

het anders kan bewijst de ontwikkeling van mathematische semantiek voor complexe talen met lastig te doorziene interpretaties zoals programmeertalen [3] en de natuurlijke taal [6] [8]. Dat het alternatief van een echte wiskundige semantiek ook bestaat voor de ontwerpers van informatiesystemen kunt U lezen in het proefschrift dat Hans Weigand [17] binnenkort hoopt te verdedigen aan de Vrije Universiteit van Amsterdam.

## 2 Het Rules technology project en de taal RL

Na deze stellingname wil ik een overzicht geven van het RL project en de wijze waarop mijn semantische visie in dit project zichtbaar is geworden. Hoewel het project ongetwijfeld vele problemen in de wereld van de conceptuele modellering niet zal kunnen oplossen hoop ik een bijdrage te leveren tot het ontwikkelen van semantische kaders waarbinnen een fundamentele vraag kan worden beantwoord: *Wat betekent het voor een informatie verwerkend systeem om correct te zijn?*

De oorsprong van de taal RL is gelegen in het IBM Research Laboratorium van San Jose, alwaar ik in het jaar 1985 gedurende 8 maanden verbonden ben geweest aan het Rules Technology project van Peter Lucas. Doelstelling van dit project was het ontwerpen van een systeem waarin een scheiding kan worden aangebracht tussen enerzijds de regels die gelden in een administratief systeem en anderzijds de applicatieprogrammatuur. Deze scheiding is van een vergelijkbaar karakter als de scheiding tussen programmatuur en gegevens die teweeg wordt gebracht door het gebruik van een gegevensbank. De naam *Regelbank (rules base)* voor de systeemcomponent waarbinnen deze regels dienen te worden opgeslagen spreekt in deze boekdelen. De scheiding zou er ook toe moeten leiden dat applicatieprogrammatuur ongewijzigd kan blijven in een wereld waar de regels voortdurend veranderen.

Binnen het rules technology project werd sterk gedacht aan toepassingen op financieel administratief gebied. Een van de vroeg ontwikkelde voorbeelden van Peter Lucas gaat over de Oostenrijkse Sociale Uitkeringen. Consequentie van deze toepassingen is dat er gerekend moet kunnen worden met getallen, die geldbedragen voorstellen. Ook logische condities spelen een rol - of bepaalde artikelen vrijgesteld zijn van BTW hangt af van logische condities waarbij recursie niet bij voorbaat kan worden uitgesloten (een artikel is vrijgesteld als het als zodanig op een lijst staat of als het onderdeel is van een vrijgesteld artikel). Voor de grote collectie feiten waarmee het systeem moet kunnen werken wordt verondersteld een goed functionerend relationeel gegevensbanken systeem aanwezig te zijn.

Binnen een goed relationeel systeem zijn de gegevens op een zodanige opgeslagen wijze dat zij van verschillende kanten kunnen worden benaderd en opgevraagd. Het dient voor de verwerkbaarheid van een vraag er niet toe te doen welke attributen gegeven zijn en welke worden gevraagd (uiteraard zal dit wel grote invloed hebben op de voor de beantwoording van de query benodigde rekentijd). Dit aspect van *reversibiliteit* wilden wij in het Rules technology project handhaven, ook indien algebraïsch of logisch beschreven regels worden toegevoegd aan de feiten in de gegevensbank. In de bestaande gegevensbanken technologie kan men berekeningen op opgehaalde gegevens uitvoeren maar slechts in een van te voren opgegeven richting. De reversibiliteit van het rekenen met algebraïsche gelijkheden treedt wel op binnen de omgeving van spreadsheets, maar die kennen weer geen koppeling met een gegevensbank. Logische regels spelen een belangrijke rol bij de deductieve gegevensbanken en het logisch programmeren, maar op die gebieden ontbrak destijds een efficiënte

koppeling met de gegevensbank als informatie cruncher. Een prototype van een dergelijke koppeling kan echter eenvoudig via vertaling worden verkregen zoals wordt aangetoond in [14].

De doelstelling van het Rules technology project kwam derhalve neer op het scheppen van een gegevensbank-achtige omgeving waarbinnen geïntegreerd gebruik kan worden gemaakt van database structuren, logische regels als in een taal als Prolog, en informatie opgeslagen in de vorm van algebraïsche gelijkheden en ongelijkheden. De aldus opgeslagen gegevens moeten reversibel opvraagbaar zijn en verwerkt kunnen worden op een relationele gegevensbank (bij voorkeur een standaard systeem).

Mijn bijdrage uit 1985 aan het project bestond uit het ontwerp van een taal (provisorisch aangeduid met de letters RL) waarin de gevraagde integratie tot stand is gebracht [15]. De betekenis van de taal is verklaard aan de hand van een relationeel model dat aansluit bij de modellen zoals die bestaan voor gewone gegevensbanken. De informatie wordt beschreven door relaties in een vrijwel normale wiskundige betekenis.

Toen ik in September 1985 vertrok uit San Jose, en ik de vraag stelde of er iets van deze taal te implementeren zou zijn, werd de verwachting uitgesproken dat dat vele jaren werk zou kosten. Ongeveer een half jaar na mijn vertrek besloot IBM het Rules technology project af te breken. Over mijn taal en het bijbehorende semantische model hield ik nog wel diverse voordrachten, oa. te Bratislava [16]. In de bijbehorende congresbijdrage kunt U een meer gedetailleerde beschrijving van de taal RL en haar semantiek aantreffen.

Ik wil mij vandaag beperken tot enkele sleutel-observaties. Het idee van de relationele semantiek is ingegeven door het feit dat zowel de relaties uit de gegevensbank als de logische predicaten via hun Tarskiaanse semantiek zich laten modelleren als normale wiskundige relaties. Hetzelfde geldt voor algebraïsche gelijkheden en ongelijkheden, indien wij deze interpreteren als de grafieken waarmee wij op de middelbare school gewerkt hebben - een visie die in de wiskunde gebruikt wordt bij de affiene algebraïsche meetkunde. Kortom, alles is een relatie.

De wereld is echter ingewikkelder, zeker wanneer we rekening houden met de wijze waarop de informatie die in verschillende regels is vervat met elkaar interfereert. Dit wordt duidelijk als wij, afhankelijk van het type relatie, letten op het effect van het geven van meer dan één regel voor een relatie. Voor relaties die uit de relationele gegevensbank afkomstig zijn is dit geen optie. Iedere relatie heeft slechts één definitie. Kijken wij naar relaties behorende bij predicaten die via logische regels zijn gedefinieerd en denken wij aan wat er gebeurt in een taal als Prolog, dan zien wij dat het geven van twee regels voor eenzelfde predicaat neerkomt op het geven van alternatieve redenen waarom het predicaat vervuld kan zijn. Het geven van meer regels krijgt als betekenis de vereniging van de bijbehorende relaties (afgezien van een mogelijke complicatie voor het geval van recursieve regels). De relaties beschreven door algebraïsche gelijk- en ongelijkheden kennen weer een ander regime: twee vergelijkingen opschrijven betekent dat aan beide vergelijkingen moet worden voldaan, hetgeen impliceert dat in deze context het na elkaar opschrijven van regels correspondeert met het vormen van de doorsnijding van de twee relaties.

In deze omstandigheden blijkt het compositionaliteitsbeginsel een uiterst nuttig heuristisch hulpmiddel, want dat spreekt uit dat wat zich semantisch verschillend gedraagt reeds syntactisch verschillend geweest moet zijn. Bij nadere studie wordt ook een syntactisch verschil zichtbaar; het is immers zo gesteld dat de relaties uit de gegevensbank, evenals de relaties die beschreven worden met behulp van logische predicaten een naam

meekrijgen. Dit geldt niet voor de relatie die ontstaat als doorsnijding van de hyperoppervlakken en halfruimten gedefinieerd door de algebraïsche gelijk- en ongelijkheden. De daardoor bepaalde relatie vervult klaarblijkelijk de rol van het model van de wereld, terwijl de benoemde relaties daarbij een assisterende rol spelen.

In RL en het semantische model voor deze taal worden de drie soorten van relaties van meet af aan syntactisch onderscheiden. Iedere soort kent schrijfwijzen, ontleend aan de wereld waaruit zij afkomstig zijn. De corresponderende regels mogen in een vrije volgorde worden opgeschreven. Bij de interpretatie worden eerst de tabulaire regels die afkomstig zijn uit de wereld van de gegevensbank geïnterpreteerd - deze regels zijn weinig meer dan directe definities. Gebruik makende van deze relaties worden de clausale regels die het formaat van Horn-clauses bezitten geïnterpreteerd, waarbij mogelijk recursieve regels worden geïnterpreteerd via een kleinste dekpunt constructie. De semantiek voor dit deel van de taal is overgenomen uit het Prolog model ontwikkeld in [14]. Als op deze wijze alle genoemde relaties bekend zijn wordt een naamloze wereld relatie verkregen als doorsnijding van alle resterende constraints, in hun interpretatie van halfruimte dan wel hyperoppervlak.

Uiteraard laat ik bij deze globale uiteenzetting zeer veel details onder het vloerkleed verdwijnen; de geïnteresseerde lezer kan meer details vinden in [16] terwijl de oerdefinitie te vinden is in [15]. In dit laatste rapport wordt overigens een nog veel uitgebreidere taal met een modulariseringsmechanisme beschreven.

Er is een keer een poging gedaan om de taal RL uit te proberen voor het doel waarvoor zij ontworpen is: het opschrijven van regels. Martje Roessingh heeft een fragment van de Universitaire aanwijzing 111 die de procedure voor het benoemen van de hoogleraar beschrijft vertaald naar RL. Het fragment van ongeveer 50 regels Nederlands expandeerde tot ongeveer 250 regels programmatext. Zij liep tevens aan tegen een aantal problemen die samenhangen met de uitdrukingskracht van de taal. Regels beschrijven vaak niet hoe de wereld in elkaar zit maar hoe de wereld in elkaar zou moeten zitten, en aangezien de wereld in dit geval bestaat uit mensenwerk kunnen wij er bijna zeker van zijn dat het feitelijke verloop van de procedure zondigt tegen de regels. Voor het weergeven van dit type deontische regels is ad-hoc de oplossing gekozen deze te modelleren via een relatie die de acties die hebben plaatsgevonden beschrijft, alsmede of deze acties correct zijn, en, indien niet, waarom niet. Dat deontische regels in informatiesystemen een belangrijke eigen rol spelen is een opmerking die men ook kan aantreffen in het werk van Hans Weigand [17]. Toen deze keuze eenmaal gemaakt was bleek al snel dat alle gedefinieerde relaties een naam hadden gekregen en dat er voor de naamloze wereldrelatie (althans in dit voorbeeld) geen rol meer was weggelegd. Voor het voorbeeldfragment verwijs ik naar [10].

### 3 Op weg naar een RL prototype

In de loop van het studiejaar 1986/87 verscheen een nieuwe deelnemer aan het RL project op het toneel in de persoon van Sieger van Denneheuvel. Als AI student had hij voor zijn afstudeerwerk een systeem ontwikkeld voor het symbolisch oplossen van algebraïsche vergelijkingen als onderdeel van het elektronische natuurkunde practicum waarmee Wielinga het leergedrag van studenten wilde onderzoeken [11]. Aangezien een vergelijkingenoplosser ook onderdeel vormde van de gereedschappen die ik te San Jose had aangetroffen voor het Rules Technology project, leek het mij nuttig de heer van Denneheuvel op deze verwant-



schap te wijzen. Enkele weken later keerde deze bij mij terug met de uitdrukkelijke wens om de taal RL te mogen implementeren - iets waarvan mij te San Jose duidelijk gemaakt was dat dit een wellicht onhaalbare klus zou zijn.

Of het een kwestie van een te zwak verweer mijnerzijds is geweest, dan wel koppigheid van de kant van mijn huidige medewerker kan ik tot op de dag van vandaag niet vaststellen. Feit is echter dat Sieger in de gelegenheid werd gesteld om het door hem begeerde project uit te voeren, en dat binnen zeer korte tijd een prototype tot stand is gekomen dat een fragment van RL implementeert. Dit prototype werkt op precies die wijze waarvan we gedroomd hadden - compilatie naar een standaard gegevensbank omgeving. Het prototype wordt beschreven in [13].

Om de werking van het prototype te beschrijven beschouwen we het soort voorbeelden dat verwerkt kan worden. Het betreft hier relaties die beschreven zijn als doorsnijding van een combinatie van relaties afkomstig uit een gegevensbank en hyperoppervlakken beschreven door algebraïsche vergelijkingen. In een naïve benadering zou men voor ieder tupel uit de gegevensbank dat aan alle relationele beperkingen voldoet de algebraïsche vergelijkingen kunnen oplossen, al dan niet met numerieke methoden. Deze oplossing is ook door andere onderzoekers gekozen.

In ons prototype wordt echter een vergelijkingenoplosser gebruikt om een symbolische oplossing te bepalen waarbij alle attributen die niet afkomstig zijn uit de gegevensbank worden uitgedrukt in attributen die dat wel zijn. Het is hierbij mogelijk dat de algebraïsche vergelijkingen aanleiding geven tot extra constraints die aan de reeds expliciet gegeven relationele selectiecriteria kunnen worden toegevoegd. Het resultaat van deze symbolische oplossing kan worden verwerkt tot een nieuwe query op een gewone gegevensbank, al dan niet uitgebreid met extra constraints, waarbij de resulterende tupels worden uitgebreid via een aantal calculates voor de uitdrukbare extra attributen.

Uiteraard moeten wij rekening houden met de situatie dat het systeem van vergelijkingen niet symbolisch kan worden opgelost op de door ons gewenste wijze. Dit kan veroorzaakt worden door een te hoge algebraïsche complexiteit. Het kan ook een kwestie zijn dat het systeem van vergelijkingen onderbepaald is in hetwelk geval het prototype een oplossing met extra vrije parameters kan produceren. Hoewel deze oplossing niet langer door de gegevensbank kan worden verwerkt, biedt zij toch nuttige informatie aan de gebruiker, die bijvoorbeeld op grond van deze informatie desgewenst een extra constraint kan toevoegen.

Het prototype geeft aldus direct toegang tot een gegevensbankenmodel. Diverse problemen uit de relationele gegevensbanken wereld komen daarom bij de verdere uitbouw van het project weer om de hoek kijken. Een van de problemen waarna thans onderzoek wordt verricht in samenwerking met Gerard Renardel de Lavalette [12] heeft betrekking op mogelijke normaalvormen voor het type relationele queries met calculates dat het prototype produceert. Voor de optimalisatie van dit soort queries is het van belang van grotere collecties queries de mogelijke gelijkwaardigheid te kunnen bepalen op grond van syntactische informatie (het volledige equivalentie probleem laat zich via een reductie tot diophantische vergelijkingen op eenvoudige wijze onbeslisbaar praten). Ook is onderzoek gaande naar een axiomatische karakterisering van de relevante equivalenties - bij dit onderzoek is naast de eerder genoemde onderzoekers ook Edith Spaan betrokken.

Het prototype heeft thans nog slechts een integratie van twee types regels bewerkstelligd: de tabulaire regels en de constraints. Gegeven de resultaten uit [14] kunnen we

logische regels toevoegen door deze te vertalen naar relationele expressies. Een interessant geval doet zich voor als er recursieve logische regels zijn. In dit geval komen wij terecht bij de problematiek van evaluatie strategieën voor recursieve queries in deductieve gegevensbanken en dat is tegenwoordig een onderzoeksthema dat vele onderzoekers trekt. Onze gast uit de Volksrepubliek China, Huang Zhisheng is een inventarisatie aan het maken van in de literatuur voorgestelde evaluatiestrategieën en hoopt daaronder een geschikte strategie aan te treffen om in ons project te gebruiken.

Een berucht probleem uit de theorie van relationele gegevensbanken is het probleem of updates door een view uitgevoerd kunnen worden, en, indien mogelijk, hoe dit moet gebeuren. In RL, waar in feite iedere uitdrukking een view voorstelt, en waarbij algebraïsche vergelijkingen voor een deel de rol van basis relaties zijn gaan innemen, komt dit probleem weer aan de orde. Sieger van Denneheuvel heeft een reeks van transformaties ontworpen en in zijn prototype geïmplementeerd die in bepaalde gevallen updates kunnen omwerken tot reeksen van updates op de gegevensbanken tabellen die het beoogde effect teweeg brengen.

Via het reeds gepubliceerde werk over RL ben ik aldus terecht gekomen bij het nog lopende onderzoek, waarvan niet duidelijk is of het operationeel bruikbare resultaten zal opleveren. Het is evenmin duidelijk of het fragment van RL dat zich thans laat compileren naar een gegevensbank zich nog essentieel zal laten uitbreiden. Zeker als men eraan denkt dat de volledige taal RL een modularisatie mechanisme omvat, zal het duidelijk zijn dat er nog een lange weg te gaan is.

Verder lijkt het mij duidelijk dat de RL wereld zozeer aansluit op het relationele gegevensbanken model dat problemen die daar niet kunnen worden opgelost voor RL ook lastig zullen blijven. Wij doen niets aan object georiënteerde gegevensbanken. Aan de oplossing van het probleem van het verwerken van beeld en geluid in een gegevensbank zullen wij vooralsnog geen bijdrage kunnen leveren. En met een ingenieurs gegevensbank gevuld met hiërarchisch gestructureerde gegevens zullen wij niet zonder meer kunnen werken; wellicht dat de strategie om het Prolog fragment uit [14] uit te breiden met niet recursieve termen, die is beschreven in de doctoraalscriptie van Bas Elbers [4] hier een opening biedt.

Ik wil hiermee het overzicht van het RL project zoals dat thans in Amsterdam wordt uitgevoerd afsluiten. Dat het project tot een werkend prototype en diverse theoretische onderzoeken aanleiding heeft gegeven, meen ik vooral te moeten toeschrijven aan het feit dat bij de ontwikkeling van de taal RL, die heeft plaatsgevonden op de wijze van droogzwemmen, een duidelijk wiskundige semantiek ten grondslag heeft gelegen aan alle ontwerp beslissingen. Hierdoor zijn een aantal denkbare fundamentele ontwerpfouten die implementatie onmogelijk hadden kunnen maken, voorkomen. Voor dit project acht ik derhalve het nut van de semantiek bewezen.

## References

- [1] Hopcroft, J.E. and Ullman, J.D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley (1979)
- [2] Barendregt, H.P., *The Lambda Calculus, Its Syntax and Semantics*, North-Holland, Studies in Logic and the Foundations of Mathematics, vol 103, revised edition (1984)

- [3] de Bakker, J.W., *Mathematical Theory of Program Correctness*, Prentice-Hall (1980)
- [4] Elbers, S.J.C., *De compilatie van willekeurige Prolog structuren in een relationele database*, Report IBM Uithoorn, IBM TR 13.206, Jun 1986
- [5] Janssen, T.M.V., *Foundations and applications of Montague grammar, Part 1: Philosophy, framework, computer science*, CWI Tract 19 (1986)
- [6] Janssen, T.M.V., *Foundations and applications of Montague grammar, Part 2: Applications to natural language*, CWI Tract 28 (1986)
- [7] Janssen, T.M.V. and van Emde Boas, P., *Some observations on compositional semantics*, in Kozen, D., ed., *Logic of programs, Proceedings 1981*, Springer Lecture notes in computer science 131 (1982) pp. 137-149
- [8] Thomason, R.T., ed., *Formal Philosophy, Selected Papers of Richard Montague*, Yale University Press (1974)
- [9] Hopcroft, J.E. and Ullman, J.D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley (1979)
- [10] Roessingh, M., *RL losgelaten op aanwijzing 111*, doctoraalscriptie UvA Oct. 1987; rep. FVI-UvA-87-18
- [11] van Denneheuveel, S.J., *A Knowledge based Lab Simulation System*, doctoraalscriptie UvA, Oct. 1986
- [12] van Denneheuveel, S.J. and Renardel de Lavalette, G.R., *RL, Normalization of database expressions involving calculations*, RUU Logic Group Preprint Series 45, Mar 1989
- [13] van Denneheuveel, S.J. and van Emde Boas, P., *Constraint solving for databases*, Proc. NAIC-88, Amsterdam, Apr. 1988, pp. 173-184
- [14] van Emde Boas-Lubsen, H. and van Emde Boas, P., *Storing and Evaluating Horn-clause rules in a Relational database*, IBM J. Research and Dev. 30 (1986) 80-92
- [15] van Emde Boas, P., *RL, a language for enhanced rule based database processing, working document*, Report IBM Research, RJ 4869 (51299) 10/10/85
- [16] van Emde Boas, P., *A semantical model for integration and modularization of rules*, Proc. MFCS 12 Bratislava, Aug. 1986, Springer Lecture notes in computer science 233 (1986) pp. 78-92
- [17] Weigand, H., *Linguistically Motivated Principles of Knowledgebases systems*, Dissertatie VUA (1989)

## The ITLI Prepublication Series

**1986**

- 86-01 The Institute of Language, Logic and Information  
 86-02 Peter van Emde Boas A Semantical Model for Integration and Modularization of Rules  
 86-03 Johan van Benthem Categorical Grammar and Lambda Calculus  
 86-04 Reinhard Muskens A Relational Formulation of the Theory of Types  
 86-05 Kenneth A. Bowen, Dick de Jongh Some Complete Logics for Branched Time, Part I Well-founded Time,  
 86-06 Johan van Benthem Logical Syntax Forward looking Operators

**1987**

- 87-01 Jeroen Groenendijk, Martin Stokhof Type shifting Rules and the Semantics of Interrogatives  
 87-02 Renate Bartsch Frame Representations and Discourse Representations  
 87-03 Jan Willem Klop, Roel de Vrijer Unique Normal Forms for Lambda Calculus with Surjective Pairing  
 87-04 Johan van Benthem Polyadic quantifiers  
 87-05 Víctor Sánchez Valencia Traditional Logicians and de Morgan's Example  
 87-06 Eleonore Oversteegen Temporal Adverbials in the Two Track Theory of Time  
 87-07 Johan van Benthem Categorical Grammar and Type Theory  
 87-08 Renate Bartsch The Construction of Properties under Perspectives  
 87-09 Herman Hendriks Type Change in Semantics: The Scope of Quantification and Coordination

**1988**

- Logic, Semantics and Philosophy of Language:*
- LP-88-01 Michiel van Lambalgen Algorithmic Information Theory  
 LP-88-02 Yde Venema Expressiveness and Completeness of an Interval Tense Logic  
 LP-88-03 Year Report 1987  
 LP-88-04 Reinhard Muskens Going partial in Montague Grammar  
 LP-88-05 Johan van Benthem Logical Constants across Varying Types  
 LP-88-06 Johan van Benthem Semantic Parallels in Natural Language and Computation  
 LP-88-07 Renate Bartsch Tenses, Aspects, and their Scopes in Discourse  
 LP-88-08 Jeroen Groenendijk, Martin Stokhof Context and Information in Dynamic Semantics  
 LP-88-09 Theo M.V. Janssen A mathematical model for the CAT framework of Eurotra  
 LP-88-10 Anneke Kleppe A Blissymbolics Translation Program
- Mathematical Logic and Foundations:*
- ML-88-01 Jaap van Oosten Lifschitz' Realizability  
 ML-88-02 M.D.G. Swaen The Arithmetical Fragment of Martin Löf's Type Theories with weak  $\Sigma$ -elimination  
 ML-88-03 Dick de Jongh, Frank Veltman Provability Logics for Relative Interpretability  
 ML-88-04 A.S. Troelstra On the Early History of Intuitionistic Logic  
 ML-88-05 A.S. Troelstra Remarks on Intuitionism and the Philosophy of Mathematics
- Computation and Complexity Theory:*
- CT-88-01 Ming Li, Paul M.B. Vitanyi Two Decades of Applied Kolmogorov Complexity  
 CT-88-02 Michiel H.M. Smid General Lower Bounds for the Partitioning of Range Trees  
 CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Maintaining Multiple Representations of Dynamic Data Structures  
 CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette Computations in Fragments of Intuitionistic Propositional Logic  
 CT-88-05 Peter van Emde Boas Machine Models and Simulations (revised version)  
 CT-88-06 Michiel H.M. Smid A Data Structure for the Union-find Problem having good Single-Operation Complexity  
 CT-88-07 Johan van Benthem Time, Logic and Computation  
 CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Multiple Representations of Dynamic Data Structures  
 CT-88-09 Theo M.V. Janssen Towards a Universal Parsing Algorithm for Functional Grammar  
 CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas Nondeterminism, Fairness and a Fundamental Analogy  
 CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas Towards implementing RL
- Other prepublications:*
- X-88-01 Marc Jumelet On Solovay's Completeness Theorem

**1989**

- Logic, Semantics and Philosophy of Language:*
- LP-89-01 Johan van Benthem The Fine-Structure of Categorical Semantics  
 LP-89-02 Jeroen Groenendijk, Martin Stokhof Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
- Mathematical Logic and Foundations:*
- LP-89-03 Yde Venema Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals  
 LP-89-04 Johan van Benthem Language in Action  
 LP-89-05 Johan van Benthem Modal Logic as a Theory of Information  
 LP-89-06 Andreja Prijatelj Intensional Lambek Calculi: Theory and Application  
 LP-89-07 Heinrich Wansing The Adequacy Problem for Sequential Propositional Logic  
 LP-89-08 Víctor Sánchez Valencia Peirce's Propositional Logic: From Algebra to Graphs
- Mathematical Logic and Foundations:*
- ML-89-01 Dick de Jongh, Albert Visser Explicit Fixed Points for Interpretability Logic  
 ML-89-02 Roel de Vrijer Extending the Lambda Calculus with Surjective Pairing is conservative  
 ML-89-03 Dick de Jongh, Franco Montagna Rosser Orderings and Free Variables  
 ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna On the Proof of Solovay's Theorem  
 ML-89-05 Rineke Verbrugge  $\Sigma$ -completeness and Bounded Arithmetic  
 ML-89-06 Michiel van Lambalgen The Axiomatization of Randomness  
 ML-89-07 Dirk Roorda Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone  
 ML-89-08 Dirk Roorda Investigations into Classical Linear Logic  
 ML-89-09 Alessandra Carbone Provable Fixed points in  $\text{ID}_0 + \Omega_1$
- Computation and Complexity Theory:*
- CT-89-01 Michiel H.M. Smid Dynamic Deferred Data Structures  
 CT-89-02 Peter van Emde Boas Machine Models and Simulations  
 CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas On Space Efficient Simulations  
 CT-89-04 Harry Buhrman, Leen Torenvliet A Comparison of Reductions on Nondeterministic Space  
 CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree A Parallel Functional Implementation of Range Queries  
 CT-89-06 H.W. Lenstra, Jr. Finding Isomorphisms between Finite Fields  
 CT-89-07 Ming Li, Paul M.B. Vitanyi A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)  
 CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet Honest Reductions, Completeness and Nondeterministic Complexity Classes  
 CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet On Adaptive Resource Bounded Computations  
 CT-89-10 Sieger van Denneheuvel The Rule Language RL/1
- Other Prepublications:*
- X-89-01 Marianne Kalsbeek An Orey Sentence for Predicative Arithmetic  
 X-89-02 G. Wagemakers New Foundations: a Survey of Quine's Set Theory  
 X-89-03 A.S. Troelstra Index of the Heyting Nachlass  
 X-89-04 Jeroen Groenendijk, Martin Stokhof Dynamic Montague Grammar, a first sketch  
 X-89-05 Maarten de Rijke The Modal Theory of Inequality  
 X-89-06 Peter van Emde Boas Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project