Voorwoord

Deze skriptie ligt op het gebied van de kunstmatige intelligentie (*Artificial Intelligence, AI*), een raakvlak van informatika en filosofie. Dit lijkt een niet erg voor de hand liggende keuze voor een student Bestuurlijke Informatika (BI), daar deze studie een raakvlak tracht te zijn van informatika en ekonomie. Toch is de kunstmatige intelligentie onderdeel van de studie Bestuurlijke Informatika en één van de drie onderzoeksterreinen van de Vakgroep Informatika van de Erasmus Universiteit.

Gegeven mijn interesse in wiskunde en filosofie kwam ik binnen de informatika 'automatisch' bij de kunstmatige intelligentie terecht. Beide interessen zijn terug te vinden in deze skriptie: het eerste gedeelte is filosofisch, het tweede (en veruit grootste) wiskundig.

De studie BI heeft veel weg van een kursus *software engineering*; de meeste skripties bevatten dan ook een implementatie van het behandelde onderwerp[1]. In eerste instantie was het ook de opzet van deze skriptie een lerend systeem te implementeren, gebaseerd op het theoretische concept *termpartities*. Er is wel onderzocht hoe de toepassing van *termpartities* tot efficiëntere algoritmen zou kunnen leiden, maar er is geen lerend systeem geïmplementeerd.

Het is binnen de informatika gebruikelijk een skriptie in het Engels te schrijven omdat het merendeel van de literatuur op dit vakgebied in het Engels is. Ik heb dat niet gedaan omdat ik had ingeschat dat het schrijven in het Engels mij teveel tijd zou gaan kosten. Achteraf gezien was een Engelse versie van (het wiskundige deel van) de skriptie toch eenvoudiger geweest.

De vertaling van de Engelse termen is soms problematisch. Ik heb getracht zoveel mogelijk woorden te vertalen, soms staat de Engelse ekwivalent tussen haakjes achter de vertaling. Bij een aantal woorden bleek een vertaling niet mogelijk omdat de Engelse woorden reeds zijn ingeburgerd in het Nederlands en in een aantal gevallen heeft de vertaling tot enigszins gezochte konstrukties geleid.

Het was de opzet dat deze skriptie een *self-contained book* zou worden, de lezer zou haar moeten kunnen lezen zonder nadere literatuur te hoeven raadplegen.

[1] Om serieus genomen te worden in de informatika moeten ideeën worden geïmplementeerd; het is zelfs zo dat deze implementatie meestal vooraf gaat aan de wiskundige analyse. Ik heb de indruk dat de meeste informatici geen goede wiskundigen zijn.

Het gevolg is dat het aantal pagina's van deze skriptie boven het gemiddelde aantal pagina's van skripties ligt en dat er soms vrij uitgebreid wordt ingegaan op voor insiders vrij triviale onderwerpen, bijvoorbeeld in het hoofdstuk over logika. Al heeft deze skriptie dan nooit een ekspliciete doelstelling gehad, zo kan zij toch nog een doel vervullen: het inleiden van onderzoekers op het gebied van induktief leren binnen een eerste-orde logika[2].

Er zijn verschillende personen die hebben bijgedragen aan de totstandkoming van deze skriptie. Als eerste wil ik mijn begeleidster Shan-Hwei Nienhuys-Cheng bedanken voor haar intensieve begeleiding en betrokkenheid bij mijn onderzoekingen. Een inspirerend en motiverend onderdeel van deze begeleiding was de organisatie van een wekelijks *seminar*, waar beurtelings iemand vertelde over zijn onderzoek. Hierbij wil ik de overige *seminar*-leden Patrick van der Laag en Mark Polman bedanken voor het beluisteren van mijn praatjes en hun eigen bijdrage aan het *seminar*.

De leesbaarheid van deze skriptie is te danken aan een aantal mensen die de vele voorgaande versies ervan hebben doorgelezen en kommentaar leverden, waarvoor ik in het bijzonder Michel Knops, Patrick van der Laag, Gert-Jan Lokhorst en Christ Leijtens wil bedanken. Christ Leijtens wil ik speciaal bedanken voor zijn hulp bij de lay-out van deze skriptie in LaTeX.

Tot slot wil ik de ontwerpers van *dream factory* bedanken voor het schrijven van het komputerspel *supaplex*, dat voor de noodzakelijke afleiding zorgde tijdens het schrijven van deze skriptie.

[2]Deze skriptie kan worden gekarakteriseerd als een overzicht van de wiskundige fundamenten en de belangrijkste ontwikkelingen op het gebied van induktief leren binnen een eerste-orde logika, of beter: als een inleiding.

Inhoud

Lijst van figuren

Hoofdstuk 1

Inleiding

"Logisch dat je afstudeert, ik begrijp niet dat je het zolang hebt volgehouden."

Deze skriptie omvat een onderzoek naar wiskundige eigenschappen van ordeningen op formules binnen een eerste-orde logika die ten grondslag liggen aan induktief lerende systemen.

Naast de analyse van ordeningen worden de induktieve modellen besproken waarin deze ordeningen worden toegepast. De motivatie van deze modellen wordt geanalyseerd aan de hand van wetenschapsfilosofische theorieën, waarvan ook de kennis- en taaltheoretische achtergronden worden behandeld.

Het probleemgebied van het onderzoek is leren binnen de logika. Zowel binnen de filosofie als binnen de informatika zijn logika en leren twee belangrijke onderzoeksgebieden. Zij hebben ieder echter een eigen kijk op logika en leren, veroorzaakt door hun verschillende doelstellingen.

In dit hoofdstuk wordt het probleemgebied behandeld, wordt de motivatie voor de aanpak van het onderzoek besproken en wordt de verdere opzet van de skriptie gegeven.

1.1 Logika

Het vakgebied logika is ontstaan binnen de filosofie en later toegepast in de informatika. Binnen de filosofie wordt logika gehanteerd om redeneringen te formaliseren, binnen de informatika om kennis in een komputer op te slaan.

1.1.1 Filosofie

Logika binnen de filosofie kent drie belangrijke stromingen: klassieke logika, induktieve logika en wiskundige logika.

Klassieke logika

In het dagelijks taalgebruik wordt vaak gebruik gemaakt van begrippen uit de logika; een uitspraak wordt logisch genoemd als hij rationeel gezien juist is.

Bij dat taalgebruik wordt verwezen naar de klassieke (traditionele) logika, de leer van het ordelijke en juiste denken. Zij houdt zich niet bezig met het denken als psychologisch verschijnsel of met de kennis-theoretische achtergrond ervan, maar met de denkpatronen, de vormen volgens welke het denken moet verlopen om tot de juiste uitkomst te geraken. Zij wordt daarom ook wel *formele logika* genoemd.

Met behulp van de klassieke logika kunnen redeneringen en gevolgtrekkingen worden beoordeeld op hun geldigheid. Een redenering wordt hierbij onderverdeeld in een aantal syllogismen, waarbij uit de waarheid van één of meer oordelen, de aannamen, wordt besloten tot de waarheid van een ander oordeel, de konklusie. De aannamen en konklusie kunnen ook worden gezien als oorzaak en gevolg, zodat kausale redeneringen in logika kunnen worden weergegeven. Logika als wetenschap is een schepping van de Griekse filosoof Aristoteles en is sindsdien een belangrijk onderzoeksgebied binnen de filosofie.

Centraal in de klassieke logika staat de overtuigingskracht van geformaliseerde redeneringen. De doelstelling is komplekse redeneringen te herleiden tot eenvoudige, onbetwistbare stappen, de syllogismen. Als de aannamen van een redenering en de geldigheid van de syllogismen worden erkend, moeten ook de redenering die uit deze syllogismen is opgebouwd en de konklusie die eruit volgt worden erkend. Er is hierbij sprake van deduktief redeneren: met noodzakelijkheid wordt gekonkludeerd tot een stelling die impliciet reeds in de aannamen bevat ligt.

Het herleiden van komplekse redeneringen naar eenvoudige syllogismen is – naast een middel voor overtuiging – ook een middel voor ontmaskering van drogredenen. Aristoteles hanteerde logika voor analyse van retorische betogen van Sofisten, Griekse wijsgeren die rondtrokken door het oude Griekenland en mensen succes in het openbare leven leerden door hen de kunst van het overtuigen bij te brengen. Plato schreef enige dialogen over de Sofisten die hen een slechte naam heeft bezorgd: sofisme is nu een synoniem voor drogreden.

Induktieve logika

De opbloei van de natuurwetenschappen sinds de zestiende eeuw wekte belangstelling voor de hierin gevolgde methode en voegde aan de logika een nieuwe tak toe, de zogenaamde *induktieve logika.* Zij onderscheidt zich van deduktieve logika doordat bij een induktieve redenering (met min of meer zekerheid) wordt gekonkludeerd tot een stelling die niet zonder meer in de aannamen ligt opgesloten.

Wiskundige logika

In het midden van de negentiende eeuw ontstond de *wiskundige* of *symbolische logika*, die zich van de traditionele logika onderscheidt door strengere axiomatische opbouw, konsekwent gebruik van symbolen in plaats van natuurlijke taal en uitbreiding van nieuwe redeneervormen.

In de wiskundige logika vervalt de doelstelling van de traditionele logika, de kleinste stappen hoeven niet meer door alle mensen direkt te worden herkend als onbetwistbare, korrekte (deduktieve) afleidingen.

De belangrijkste doelstelling van de wiskundige logika was het formaliseren van de getaltheorie van de wiskunde in de logika. De getaltheorie komt ons zeer vanzelfsprekend en intuïtief korrekt voor, maar binnen de filosofie wordt zij door velen gezien als een menselijke schepping zonder waterdichte fundering.[1] Zo vraagt men zich af of het mogelijk is tot oneindig te tellen. Aan het begin van deze eeuw werd getracht een fundering te geven met behulp van oer-oude en intuïtieve principes uit de logika. Gödel toonde in de jaren dertig van deze eeuw aan dat dit onmogelijk is.

1.1.2 Informatika

Wiskundige logika staat aan de basis van een representatieformalisme binnen de kennisrepresentatie, een vakgebied binnen de informatika dat onderzoek uitvoert naar representatie en manipulatie van kennis met behulp van een komputer. Deze kennis wordt opgeslagen in een zogenaamd kennissysteem (*knowledge based system*).

De doelstelling van de logika als representatieformalisme is niet het herleiden van komplekse redeneringen of het afleiden van de wiskundige getaltheorie, maar het efficiënt formaliseren en representeren van kennis. Het gaat niet om het ontdekken van kennis maar om het formaliseren ervan, daarom wordt er ook wel gesproken over informatierepresentatie.

Twee logika's

De propositielogika en de eerste-orde predikatenlogika zijn twee bekende logika's die veel worden gebruikt als kennisrepresentatieformalisme. Het verschil tussen beide is dat de eerste-orde predikatenlogika een uitbreiding op de propositielogika is en relationele kennis kan representeren, bijvoorbeeld temporele redeneringen, planningsproblemen en kwalitatieve redeneringen.

Algemeen wordt aangenomen dat het uitdrukkingsvermogen van de propositielogika te beperkt is en dat het uitdrukkingsvermogen van een algemene AI-taal minstens zo groot zou moeten zijn als het uitdrukkingsvermogen van de eersteorde predikatenlogika [13].

[1] Deze reflektie op de fundamenten van de wiskunde heeft geleid tot belangrijke ontdekkingen, bijvoorbeeld de niet-Euclidische meetkunden binnen de wiskunde en de kwantum-mechanika binnen de natuurkunde.

Prolog

Logika heeft zijn populariteit als representatieformalisme voor een belangrijk deel te danken aan de programmeertaal Prolog, gebaseerd op Horn-clause logika, een deelverzameling van de eerste-orde predikatenlogika.

Een opmerkelijk verschijnsel in Prolog is dat de meeste concepten rekursief (naar zichzelf verwijzend) worden gedefinieerd. Dit is naar mijn mening opmerkelijk daar de mens dit slechts in zeer beperkte mate schijnt te doen. De doelstelling van de AI, het simuleren van menselijke redeneringen, is met Horn-clause logika dan ook niet te bereiken.

1.2 Leren

Filosofie en informatika leveren twee verschillende invalshoeken voor leren. Binnen de filosofie is leren een noodzakelijke voorwaarde om intelligent genoemd te mogen worden en wordt leren beoordeeld op kennistheoretische gronden. In de informatika worden leerprincipes toegepast om de mogelijkheden van kennissystemen uit te breiden.

1.2.1 Filosofie

De kunstmatige intelligentie houdt zich bezig met het bouwen van geautomatiseerde systemen die de menselijke intelligentie simuleren. Een belangrijke filosofische vraag is wat die menselijke intelligentie eigenlijk is. Hierover bestaat binnen de filosofie geen konsensus, maar men is het er over eens dat leren een noodzakelijke voorwaarde is voor een systeem om intelligent genoemd te mogen worden.

Leren is een algemene term waar verschillende soorten processen onder vallen. Wij leren (onthouden) vermenigvuldigingstafels, leren (ontdekken hoe) te lopen, leren (begrijpen en kunnen toepassen) een taal. Binnen de (wetenschaps-)filosofie is leren de groei van kennis.

Leren is kompleks omdat er vaak veel kennis bij betrokken is. Een voorbeeld hiervan is het probleem van de wandelende bladeren. Een boom heeft lichte en donkere bladeren, afhankelijk van de stand van de zon. Aangezien de positie van de lichte bladeren gedurende de dag wijzigt zou kunnen worden gekonkludeerd dat de bladeren zelf van positie wijzigen. Dat dit geen korrekte konklusie is kan slechts worden bewezen als er kennis beschikbaar is over de manier waarop bomen groeien, hoe de zon gedurende de dag van positie verandert en wat de invloed van zonlicht op de kleur van objekten is.

Bij de definitie van leren – groei van kennis – kunnen twee vragen worden gesteld: wat is kennis en wat is een goede methode van leren.

Het antwoord van het logisch positivisme op deze vragen was gebaseerd op induktieve logika. Popper stelde daartegenover dat induktie overbodig is in de wetenschap, hoe kennis wordt afgeleid is niet van belang.

1.2.2 Informatika

Machinaal leren (*machine learning*) is een onderzoeksgebied binnen informatika waarin leermethoden worden bestudeerd. Binnen logika zijn deze leermethoden gebaseerd op induktie.

Leren is kompleks omdat het minder algemeen te formuleren is dan deduktieve afleidingen; bij leren is de betekenis bijna altijd van belang. Bij deduktie volgt de konklusie met zekerheid uit de aannamen, bij induktie is het nooit zeker of de geïnduceerde uitspraak korrekt is. Het probleem van de wandelende bladeren is hier een voorbeeld van: voor de komputer ligt het voor de hand te konkluderen dat de positie van de bladeren verandert.

Logika's

Induktie werd eerst toegepast bij de bouw en onderhoud van kennissystemen, gebaseerd op propositielogika. Dit leverde een aanzienlijke tijdsbesparing op.

Omdat de propositielogika een beperkt uitdrukkingsvermogen had, werd in de jaren zeventig getracht de induktieve systemen op de eerste-orde predikatenlogika te baseren. Het duurde tot het einde van de jaren tachtig voor hier de eerste successen werden behaald.

Toepassingen

Induktieve leerprincipes worden binnen een eerste-orde logika op drie gebieden toegepast: het ontwikkelen van kennissystemen, het uitbreiden van kennis in kennissystemen en het opsporen van tegenspraken in kennissystemen.

De belangrijkste toepassing is het ontwikkelen van kennissystemen. Gezien de kompleksiteit van induktie zijn aan deze systemen – naast het induktiemechanisme – verschillende andere mechanismen toegevoegd.

Een voorbeeld van een kennissysteem dat met behulp van induktie is gebouwd is `BMT`, een systeem dat de konfiguratie van vuurbeschermende uitrustingen in gebouwen bepaalt. Het systeem is in negen jaar ontwikkeld en bevat 30.000 regels, waarmee het (in 1990) ruimschoots het grootste kennissysteem is dat voltijds commercieel wordt gebruikt [32].

Een andere toepassing is het uitbreiden van inkomplete kennissystemen [63]. In plaats van een heel kennissysteem met behulp van induktie op te bouwen wordt aan een reeds bestaand kennissysteem kennis toegevoegd, hetgeen minder kompleks is dan het bouwen van een heel kennissysteem omdat het aantal mogelijke hypothesen veel kleiner is. Er wordt wel gesteld dat minstens negentig procent van de kennis reeds geformaliseerd moet zijn voor effektief induktie kan worden toegepast.

Een voorbeeld van deze toepassing is de uitbreiding van kennis van een juridisch kennissysteem, waarin de regels van het wetboek zijn vastgelegd. Naast het wetboek maken rechters ook gebruik van jurisprudentie, die de rechtsopvattingen van een land weergeeft en een aanvulling op het wetboek is. In de jaren tachtig bestonden er bijvoorbeeld geen wetregels voor bescherming van software, maar er waren wel enkele arresten en vonnissen op dit gebied. Een juridisch kennissysteem kan deze arresten (induktief) generaliseren zodat het systeem ook op het gebied van software-bescherming uitspraken kan doen.

De derde toepassing is het opsporen van tegenspraken in (*debuggen* van) kennissystemen [57]. Bij de vorige twee toepassingen wordt ervan uitgegaan dat de reeds aanwezige kennis geen fouten bevat, een in de praktijk niet erg realistische aanname. De fouten kunnen zijn veroorzaakt door foutieve invoer van kennis, doordat (induktief) geleerde veronderstellingen niet juist zijn of doordat de kennis veranderd is.

Een voorbeeld hiervan zijn bureaukratische systemen waarin voorschriften zijn geformaliseerd. Deze voorschriften kunnen in een specifiek geval leiden tot tegengestelde orders. De leerprincipes kunnen in zo'n systeem worden toegepast om deze tegenspraken op te sporen en te bepalen welk voorschrift domineert.

1.3 Motivatie van de gekozen aanpak

De gebruikelijke aanpak van onderzoek op het gebied van induktief leren is een eksperimentele: eerst wordt een induktief lerend systeem gebouwd en vervolgens worden de eigenschappen van dit systeem geanalyseerd.

In eerste instantie was het ook de opzet van deze skriptie een lerend systeem te implementeren, gebaseerd op het theoretische concept *termpartities*.

Ik heb daarom enige eenvoudige basisoperaties van *termpartities* geïmplementeerd in de programmeertaal Pascal en later heb ik onderzocht hoe een lerend systeem zou kunnen worden geïmplementeerd in Smalltalk-V, dat als voordeel heeft dat het een Prolog-implementatie (Prolog-V) bevat. Ik geloof dat Prolog als enige programmeertaal een aantal nadelen heeft; de kombinatie van Smalltalk en Prolog zou een goede basis kunnen zijn voor een lerend systeem[2].

Er is geen lerend systeem geïmplementeerd. Dit is gedeeltelijk veroorzaakt door tijdgebrek, maar hoofdzakelijk door mijn interesse in wiskundige problemen op het gebied van induktief leren. In deze skriptie beperk ik mij tot enige aanwijzingen hoe het gebruik van *termpartities* tot efficiëntere algoritmen zou kunnen leiden.

De opzet van het onderzoek in deze skriptie is filosofisch en wiskundig.

[2] Prolog en Smalltalk zijn programmeertalen, gebaseerd op respektievelijk resolutie (backtracking) en frames (overerving, data-hiding). In de praktijk van het programmeren is het gebruik van deze concepten wel eens handig, door hun volgelingen zijn zij echter tot religie uitgeroepen. Pragmatisch gezien kan worden gesteld: het moet geen doel op zich worden [61].

Er zijn zowel pogingen gedaan om de logika-concepten in een frame-omgeving in te bouwen (als in Smalltalk-V) als frame-concepten in een logika-omgeving [60]. Persoonlijk geef ik de voorkeur aan de eerste aanpak.

1.3.1 Filosofie

Er zijn drie redenen voor de behandeling van de filosofische theorieën: analyse van de motivatie van induktieve modellen, vaststelling van een kader voor induktief leren en mijn persoonlijke interesse.

De eerste reden voor een filosofisch onderzoek is de motivatie door wetenschapsfilosofische theorieën van de induktieve modellen van Plotkin [48], Shapiro [56] en Muggleton [32]. De wetenschapsfilosofische theorieën zijn op hun beurt gefundeerd door kennis- en taaltheorieën.

Een andere reden om de filosofische achtergronden te behandelen is het geven van een kader waarbinnen gewerkt kan worden. De stromingen binnen de kunstmatige intelligentie en binnen de kennisrepresentatie worden behandeld en de relatie met de filosofische theorie wordt besproken.

De ontwikkelingen in het relatief jonge vakgebied van de kunstmatige intelligentie zijn sterk beïnvloed door filosofische theorieën van intelligentie en kennis. De ontwikkelingen binnen de AI hebben eveneens invloed op het filosofische denken, bijvoorbeeld op de filosofische antropologie:

> By promising (or threatening) to replace man, the computer is giving us a new definition of man, as an 'information processor', and of nature as 'information to be processed'. I call those who accept this view of man and nature Turing's man. – J.D. Bolter [6, p. 13]

Het begrijpen van een natuurlijke taal is het meest kenmerkende van onze menselijke intelligentie en een noodzakelijke voorwaarde voor een geautomatiseerd systeem om intelligent genoemd te mogen worden. Volgens de AI-pioneer Alan Turing is het zelfs een voldoende voorwaarde om intelligente systemen te kunnen onderscheiden van niet-intelligente [5, p. 40].

De laatste reden is een persoonlijke; filosofie fascineert mij. Ik ben van mening dat onderzoekers in de wetenschap zich bewust moeten zijn van de (filosofische) kontekst van hun werkzaamheden en de mogelijke gevolgen van hun onderzoek[3].

Wetenschap moet worden gerechtvaardigd als maatschappelijk goed. Het is hierbij de vraag wat haar praktische nut is en of de waarheid zelf een specifieke kulturele waarde heeft.

> Wie – behalve een paar grote kinderen die in met name de natuurwetenschappen te vinden zijn – gelooft momenteel nog dat kennis van de astronomie, biologie, fysica of chemie ons iets kan vertellen over de zin van de wereld, of zelfs maar over de weg waarlangs die zin – gesteld dat hij bestaat – te vinden zal zijn? – Max Weber [62, p. 129]

[3] Het draait hier dus niet om de toepassing van de wetenschap maar om de wetenschap zelf. In het geval van kennissystemen heeft niet alleen de toepassing van een kennissysteem, maar ook het algemene ontwerp van een kennissysteem morele implikaties.

1.3.2 Wiskunde

Onderzoekers op het gebied van kunstmatige intelligentie kunnen worden onderverdeeld in *eksperimentalisten* en *formalisten* [13]. Formalisten beweren dat eksperimentalisten meer vorderingen zouden maken als zij meer kennis van de verschillende theoretische ideeën zouden hebben; eksperimentalisten beweren daartegenover dat formalisten behulpzamer zouden zijn als zij zich minder met vorm en meer met inhoud bezig zouden houden.

De meeste ontwikkelingen binnen kunstmatige intelligentie (en binnen andere wetenschappen) zijn geïnspireerd door eksperimentalisten. Formalisten dienen hoofdzakelijk om orde te scheppen in het eksperimentele werk. Ook het meeste onderzoek op het gebied van induktief leren is gericht op werkende systemen. Er wordt uitgegaan van een implementatie van een lerend systeem, vervolgens wordt het systeem min of meer wiskundig beschreven. De relatie tussen de vele verschillende systemen is echter in beperkte mate geanalyseerd. Deze skriptie is een bijdrage aan het formele raamwerk, een 'ordening' in het werk van eksperimentalisten.

Wetenschappelijke vooruitgang heeft een wiskundig apparaat nodig om goede ideeën in uit te drukken en te unificeren. De theoretische en praktische arbeid dragen gezamenlijk bij aan de wetenschappelijke vooruitgang.

Belangrijke nieuwe resultaten binnen de kunstmatige intelligentie zouden echter gelanceerd moeten worden vanaf een solide theoretisch platform. Met Popper stel ik dat het niet de taak van de eksperimentalisten is nieuwe ideeën te initialiseren, maar te trachten de formele theorieën te onderbouwen of te falsificeren.

Van de wiskunde staat in deze skriptie de logika centraal. Hoewel het logikabolwerk veel te duchten heeft gehad van aanvallen van andere kennisrepresentatieformalismen, is zij nog steeds een belangrijk onderzoeksgebied.

Logika is een deklaratief representatieformalisme, dit betekent dat kennis wordt gerepresenteerd met feiten en dat afleidingen identiek zijn (onafhankelijk van de betekenis). Dit concept heeft een lange traditie binnen de filosofie en is onderhevig aan veel (terechte) kritiek. Binnen de informatika zijn er echter problemen waarvoor deklaratieve formalismen de beste oplossing zijn.

Een belangrijk voordeel van deklaratieve formalismen is de mogelijkheid bij redeneren – naast de konklusie – ook de redenering weer te kunnen geven. Bij induktief leren kan worden verantwoord hoe de geleerde regels tot stand zijn gekomen. In veel toepassingen is deze verantwoording belangrijk, bijvoorbeeld in een juridisch systeem dat wordt gebruikt als adviseur bij rechtzaken. Het is dan niet zozeer van belang welke uitspraak het systeem doet, maar via welke wetregels deze uitspraak tot stand is gekomen.

1.4 Verdere indeling skriptie

In hoofdstuk 2 worden de ontwikkelingen binnen kunstmatige intelligentie, kennisrepresentatie en induktieve leermodellen beschreven. Bovendien worden van deze drie gebieden de filosofische motivaties en achtergronden behandeld.

Hoofdstuk 3 tot en met 6 bestaan uit een formele inleiding in de gehanteerde logika's en een wiskundige analyse van ordeningen die ten grondslag liggen aan de induktieve modellen.

Hoofdstuk 7 behandelt de toepassing van deze ordeningen in induktief lerende systemen.

In hoofdstuk 8 staat een samenvatting van het onderzoek en worden enige onderwerpen voor verder onderzoek aangegeven.

In de appendix wordt een formele analysemethode beschreven, de logika van de logische strukturen, die gehanteerd is bij de analyse van de ordeningen. Dit is een tweede-orde logika, een logika waarin kan worden geredeneerd over de eerste-orde predikatenlogika.

Hoofdstuk 2

Filosofie en AI

Ik weet dat ik niets weet.
– Socrates

2.1 Inleiding

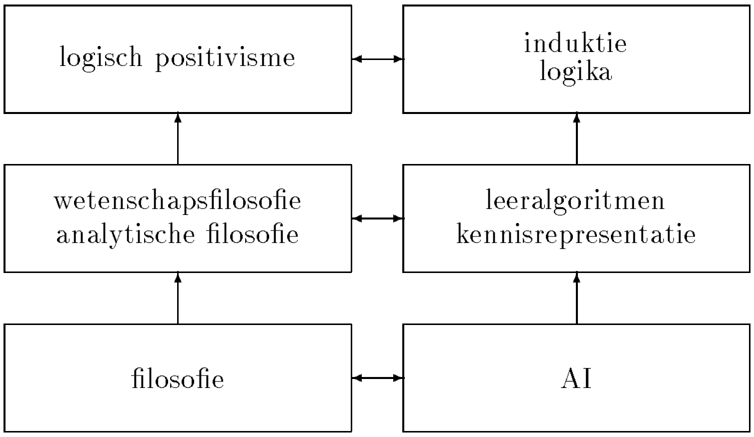
Er is reeds vele eeuwen nagedacht over menselijke kennis en intelligentie. Deze reflektie begon in de Griekse oudheid met Socrates, die de mens zag als een intelligente en morele persoonlijkheid. In Europa ontstond voor het eerst het idee dat de mens een bewustzijn heeft en onderscheid kan maken tussen goed en kwaad, een idee dat door zijn leerling Plato en later door andere Griekse filosofen werd overgenomen.

Aristoteles stelde vervolgens dat de mens de verschijnselen in de wereld, zoals hij die waarneemt met zijn zintuigen, kan verklaren met behulp van zijn intelligentie. Hij stelde regels op waaraan deze verklaringen moeten voldoen. Dit systeem van regels is de voorloper van de logika.

De theorie van Aristoteles was tot de moderne tijd de enige standaard in het streven naar kennis omtrent natuur, wereld en mens (binnen wetenschap en filosofie). In de periode van de Verlichting kwam daarin weer beweging. De positie van de mens als intelligent wezen werd nog meer benadrukt: de mens kan de werkelijkheid niet alleen kennen, hij is bovendien het middelpunt van de wereld en bepaalt zelf wat kennis is.

Om in kunstmatige intelligentie menselijke intelligentie te kunnen simuleren zal eerst de vraag beantwoord moeten worden wat menselijke intelligentie eigenlijk is. Er bestaat dus een relatie tussen kunstmatige intelligentie en filosofie. Dit blijkt tevens uit het feit dat AI-theorieën sterk beïnvloed zijn door theorieën die in de filosofie zijn opgesteld.

Over kennis is binnen de filosofie vooral de laatste twee eeuwen nagedacht binnen de wetenschapsfilosofie. Voor het opstellen van leeralgoritmen (binnen een kennisrepresentatie) is veel overgenomen uit de wetenschapsfilosofie en induktie (binnen de logika) in het bijzonder heeft een sterke band met de filosofische stroming *het logisch positivisme*. In figuur 2.1 wordt dit verband weergegeven.



Figuur 2.1: Positie van logika binnen AI en filosofie

2.2 Filosofie

2.2.1 Kennistheorie

Er zijn veel filosofen die zich de afgelopen twee eeuwen met kennistheorie hebben beziggehouden. Er zijn enkele stromingen ontstaan met radikaal verschillende visies op het begrip *kennis*.

Het begrip kennis kent verschillende aspekten, die door een algemene kennistheorie moeten worden beschreven. Kennen is *het op de hoogte zijn van het bestaan van iets of iemand* [10, p. 620]. Kennen is dus een relatie met de werkelijkheid; de relatie van wat wij kennen met de werkelijkheid is het eerste aspekt dat zal moeten worden beschreven. Het tweede aspekt is het doen van uitspraken over kennis, dit gebeurt meestal in een natuurlijke taal maar soms in een wiskundige of logische taal. Een ander belangrijk aspekt is het ontstaan en wijzigen van kennis. Een algemene kennistheorie moet in ieder geval antwoord kunnen geven op de volgende vragen:

- Wat bestaat er?
- Wat kunnen wij kennen (weten)?
- Wat kunnen wij beweren?
- Hoe kunnen wij leren?

Kennen

De scheiding tussen wat er bestaat en wat wij kunnen kennen werd aan het einde van de 18e eeuw aangebracht door de verlichtingsfilosoof Immanuel Kant in zijn beroemde *Kritik der Reinen Vernunft*. Het was een revolutie in het denken en wordt wel een Copernicaanse omwenteling genoemd[1]. Deze scheiding betekent dat er een *Ding an sich* bestaat dat niet kenbaar is. De eigen struktuur van het denken is het funderende kader van het kenproces.

Alles wat wij kunnen kennen, het kennisdomein, kan in twee delen worden gesplitst: het *weten dat* en het *weten hoe*.

Het *weten dat* van het kennisdomein zijn de feiten en overtuigingen[2]. Een feit is een gedachte waarvan het, binnen een bepaald kader, vaststaat dat het overeenkomt met de werkelijkheid en een overtuiging is een gedachte waarvan wordt vermoed dat hij overeenkomt met de werkelijkheid.

Het *weten hoe* zijn heuristieken. Een heuristiek is een techniek die de efficiëntie van een zoekproces verbetert, eventueel onder opoffering van claims met betrekking tot volledigheid [50, p. 75]. Heuristieken zijn hoofdzakelijk ervaringsregels. Zo zal een schaakgrootmeester zijn kennis vooral baseren op heuristieken, zelden op volledige informatie (bijvoorbeeld in het eindspel).

Beweren

Wat wij kunnen beweren is afhankelijk van de taal waarin wij onze gedachten uitdrukken. Deze taal kan een natuurlijke taal zijn (bijvoorbeeld het Nederlands) maar ook een wiskundige of logische taal. In de negentiende eeuw kon men in de toen gebruikte natuurlijke taal niet beweren wat de atoomtheorie inhield. Dit werd toen eerst in wiskundige termen beschreven. Later werd de natuurlijke taal uitgebreid met woorden als *elektronen* en *quarks* zodat deze theorie er wel in kan worden uitgedrukt. De scheiding tussen wat wij kunnen kennen en wat wij kunnen beweren is dus afhankelijk van de gebruikte taal waarin iets wordt beweerd.

Iets wat wij beweren, een uitspraak, is een representatie van kennis. De taal waarin uitspraken worden gedaan levert alle mogelijke kennisrepresentaties. Dit wordt het uitdrukkingsvermogen van de taal genoemd. Er bestaat dus een tweedeling in kennis en representatie van kennis met relaties tussen beide.

Twee vragen kunnen bij deze tweedeling worden gesteld. De eerste vraag is of wij alles wat wij kunnen kennen ook op één of andere manier kunnen beweren, ofwel, is er een taal waarin wij alle mogelijke gedachten kunnen weergeven. De tweede is hoe kompleks de relatie tussen kennis en representatie ervan is. Is er een één-op-één afbeelding tussen uitspraken en gedachten (afbeeldingstheorie) of hangt een uitspraak altijd samen met de kontekst waarbinnen de uitspraak wordt gedaan (holisme). In de volgende paragraaf zullen deze vragen uitgebreid worden besproken.

---

[1] Copernicus stelde (reeds in de zestiende eeuw) dat de aarde niet het onbeweeglijke middelpunt van het heelal is, dus dat de eigenschappen van de sterren niet alleen afhankelijk zijn van hun schijnbare positie ten opzichte van de aarde maar dat deze eigenschappen een effekt zijn van de plaats van onze positie op de aarde. Analoog geldt voor het kennen dat wij de kenobjekten nooit objektief kunnen kennen zoals zij zijn, maar slechts zoals zij zich aan ons voordoen.

[2] Dit is de algemeen gebruikte vertaling van het Engelse *belief*. Een betere vertaling is *vermoeden*.

Verscheidene anarchistische wetenschapsfilosofen hebben de eerste vraag omgedraaid en stellen dat wij slechts kunnen kennen wat wij kunnen beweren en dat verschillende talen verschillende werkelijkheden creëren. De psycholoog Lacan verwoordde deze theorie door te stellen dat het onbewuste gestruktureerd is als een taal [22].

Een zwakke variant hiervan is het linguïstisch relativisme, waarin wordt gesteld dat de kijk op de werkelijkheid van verschillende (families van) talen dermate verschillend is dat er in de ene taal niet over de andere kan worden gesproken[3]. Zo kan in een West-Europese taal niet over de taal van de Hopi (een indianenstam) worden gesproken omdat met beide talen een andere manier van denken en een ander wereldbeeld korrespondeert. Deze stellingname, bekend als de 'Sapir-Whorf these', heeft tegenwoordig niet zo'n grote aanhang meer.

Leren

Hoe wij kunnen leren is afhankelijk van de visie op kennis en taal. Met leren wordt niet het opnemen van bekende kennis bedoeld, zoals bijvoorbeeld op school, maar het ontwikkelen van nieuwe theorieën zoals in de wetenschap. Het gaat dus om het leren van gemeenschappen, niet van individuen.

De westerse wetenschappen zijn sterk beïnvloed door het *empirisme*, de opvatting dat alle betrouwbare kennis op ervaring berust. In paragraaf 2.2.3 zullen de drie belangrijkste stromingen in de wetenschapsfilosofie worden besproken.

Andere stromingen

De belangrijkste tegenhanger van het empirisme is het *idealisme*. Voor de Griekse wijsgeren Socrates en Plato was de enige werkelijke kennis op begrippen gebaseerd; deze begrippen zijn meta-fysische ideeën, Plato's leer wordt ook wel de ideeënleer genoemd. De ethiek staat centraal in hun filosofie: kennis is gebaseerd op inzicht van het goede, het goede in de mens volgt direkt uit kennis van het goede en het kwaad uit onwetendheid. Alles wat uit zintuiglijke waarnemingen wordt afgeleid is te dubbelzinnig om "kennis" te mogen worden genoemd, Plato noemt dit *afgeleide kennis*.

Dit idealisme loopt als een rode draad door de westerse filosofie en vindt een hoogtepunt in de filosofie van Hegel, een opvolger van Kant. Kennis heeft volgens hem een drieledig karakter: de zintuiglijke waarneming, de subjektieve indruk ervan op de mens en de zelfkennis. Bij zelfkennis valt het onderscheid tussen waarnemer en waargenomene weg. De rede leidt ons, via de dialektische methode, naar de absolute idee, het absolute kennen. De dialektiek vinden we in de twintigste eeuw in een praktische vorm terug in het marxisme.

Er zijn verschillende andere belangrijke stromingen in de twintigste eeuw, die hier voor de volledigheid worden genoemd: het existentialisme, de fenomenologie en het differentiedenken. In het existentialisme wordt het bestaan, de existentie benadrukt, in de fenomenologie zijn de fenomenen het uitgangspunt voor een

[3] Vergelijk het Franse spreekwoord: *Traduction est trahison* [49].

analyse van de werkelijkheid. Het differentiedenken is een tegenhanger van de dialektiek, in plaats van de tegenstelling wordt het verschil als paradigma gehanteerd. De differentiedenkers zijn sterk beïnvloed door Friedrich Nietzsche, een radikaal filosoof met een literaire stijl van filosoferen. Tot slot een citaat van hem over de relatie tussen kennis (gedachten) en ervaringen:

> Gedachten zijn de schaduwen van onze ervaringen – altijd donkerder, leger, simpeler dan deze. – Friedrich Nietzsche [45, p. 147]

2.2.2 Taaltheorie

De natuurlijke taal is van oudsher het medium waarin wij mensen iets beweren. Het is een formalisme voor kennisrepresentatie, maar het uitdrukkingsvermogen en de relatie met de kenbare werkelijkheid is niet duidelijk. Om dit te verduidelijken wordt gezocht naar ekwivalente representatieformalismen.

De natuurlijke taal is redundant en ambigu. Verschillende uitspraken kunnen hetzelfde betekenen en één uitspraak kan verschillende betekenissen hebben: er bestaat een veel-veel relatie tussen kennis en representaties ervan in de natuurlijke taal. Er is daarom ook een praktische reden om te zoeken naar betere representatiemethoden met dezelfde uitdrukkingskracht. Een efficiënt en eenduidig representatieformalisme zou de (wetenschappelijke) kommunikatie aanzienlijk kunnen verbeteren.

Atomisme

De logika lijkt een voor de hand liggend formalisme om kennis en gevolgtrekkingen in weer te geven. De idealist Kant schreef reeds:

> Het domein van het kenbare kan worden beschreven met behulp van algemene principes en kategorieën die in hun zuiverste vorm in de logika worden gevonden. [22]

De zinnen "Gras is groen", "De Erasmus Universiteit is groot" en "Stelen is slecht" hebben alle de logische struktuur "... is ...". Zinnen als "Alle mensen zijn sterfelijk", "Alle kinderen moeten naar school" en "Op iedere regel is een uitzondering" hebben eveneens dezelfde logische struktuur en kunnen in een logika worden geformaliseerd als "voor alle ... geldt ...".

Russell en Wittgenstein werkten in het begin van de twintigste eeuw samen aan een logische taal om uitspraken binnen de natuurlijke taal weer te geven, het logisch atomisme. Russell zocht naar een ideale logische taal om de uitspraken van de natuurlijke taal in uit te drukken, Wittgenstein beweerde dat vanuit het gezichtspunt van de logika gezien alle talen hetzelfde zijn, er is maar één *Sprachlogik*. Deze taallogika is echter niet af te leiden uit de natuurlijke talen. De echte logische vorm kan niet worden beschreven, alleen getoond:

> 4.1212. Was gezeigt werden *kann*, *kann* nicht gesagt werden. – Ludwig Wittgenstein [64, p. 54]

Slechts in de toepassing van een natuurlijke taal kan blijken of de taallogika wordt gevolgd of overtreden. De taal zelf is logisch volmaakt in orde.

Zinnen in de taal zijn uiteindelijk te herleiden tot uitspraken over logische atomen. Deze logische atomen verwijzen volgens Russell naar zintuiglijke waarnemingen (*sense-data*). Volgens Wittgenstein zijn deze atomen namen van objekten, die kunnen worden geleerd door *ostensie* ('aanwijzing'), de betekenis kan wederom slechts getoond worden in het gebruik van een taal.

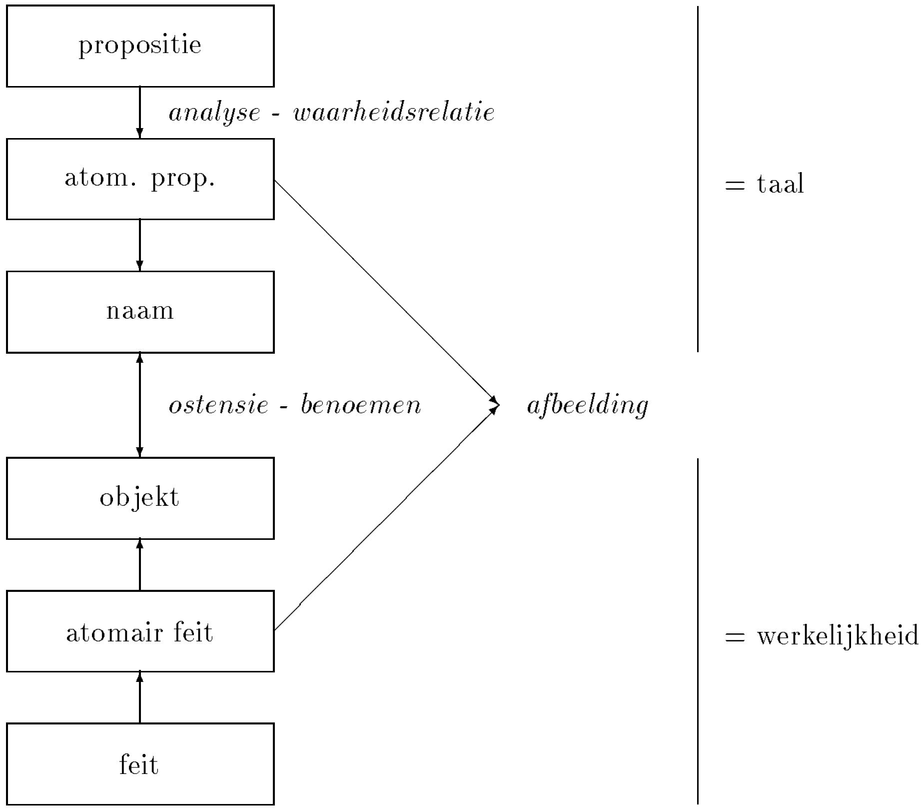
Sommige zinnen moeten eerst worden herschreven voor ze in een formele taal kunnen worden gerepresenteerd. Zo kan bijvoorbeeld over iets wat niet bestaat ook geen uitspraak worden gedaan. Volgens Russell moeten uitspraken over zogenaamde *bepaalde beschrijvingen* worden herschreven naar uitspraken over *sense-data*. De zin "A is X" wordt herschreven naar "Er is in het universum een A en deze A heeft de eigenschap X". De zin "De koning van Frankrijk is kaal" is zo te herschrijven naar "Er is een koning van Frankrijk en die koning is kaal". De ontkenning van deze zin, "Het is niet zo dat de koning van Frankrijk kaal is", blijkt nu op twee verschillende manieren herschreven te kunnen worden, namelijk als "Er is geen koning van Frankrijk" en "Er is een koning van Frankrijk en deze koning is niet kaal". Hieruit blijkt de ambiguïteit van de natuurlijke taal. Russell liet zien dat de schijnbare logische vorm niet de echte logische vorm hoeft te zijn.

Wittgenstein stelde een afbeeldingstheorie op. Deze theorie gaat ervan uit dat de taal een afbeelding van de werkelijkheid geeft. Dit betekent dat er een overeenkomst bestaat tussen de struktuur van de taal en de struktuur van de werkelijkheid. De vorm van een uitspraak moet gelijk zijn aan de vorm van de werkelijkheid, als dit niet het geval is dan is de uitspraak niet betekenisvol:

> Was sich überhaupt sagen läßt, läßt sich klar sagen; und wovon man nicht reden kann, darüber muß man schweigen. – Ludwig Wittgenstein [64, p. 8]

De logische taal bestaat volgens de afbeeldingstheorie uit atomaire proposities en samengestelde proposities. De atomaire proposities zijn direkt verbonden met de wereld, hun waarheidswaarde wordt niet door andere proposities maar slechts door de wereld bepaald. Samengestelde proposities kunnen worden begrepen uit de atomaire proposities. In de taal drukken wij onze gedachten uit die plaatjes zijn van de werkelijkheid. Deze plaatjes kunnen kloppen of niet kloppen en de uitspraken kunnen waar of niet waar zijn. De uitspraken en de gedachten zijn beide feiten, die vergeleken kunnen worden. De afbeeldingstheorie stelt dat zo'n vergelijking altijd mogelijk is. Dit geldt bijvoorbeeld voor een muziekstuk, een grammofoonplaat ervan, de partituur, de fysische geluidsgolven, zij hebben alle dezelfde logische struktuur. Het schema in figuur 2.2 laat zien dat de taal in de afbeeldingstheorie een spiegelbeeld is van de werkelijkheid.

De natuurlijke taal bestaat uit uitdrukkingen die met behulp van logische konstruktiemiddelen in de taallogika kunnen worden weergegeven, dus zonder dat daarbij conceptualisering een rol speelt. Zo kan de uitspraak "een vrijgezel is ongehuwd" door middel van de definitie "een vrijgezel is een ongehuwd huwbaar



Figuur 2.2: De afbeeldingstheorie

persoon" omgezet worden in de logische tautologie "een ongehuwd huwbaar persoon is ongehuwd". Alle uitspraken kunnen op deze manier worden herschreven.

De taallogika bestaat uit twee delen, een conceptueel raamwerk bestaande uit logika met synoniemdefinities en inhoudelijke kennis. Inhoudelijke kennis wordt geformuleerd in de door het conceptuele schema geformuleerde taal. Deze inhoudelijke kennis is een reduktie van alle kennis en bestaat uit zintuiglijke indrukken (Russell) of atomen (Wittgenstein).

Holisme

De atomistische visie op het wezen van de taal is later door Wittgenstein zelf volledig onderuit gehaald. Zijn kritiek is tweeledig. Ten eerste is taal vaak iets anders dan benoemen en beschrijven. Ten tweede kan de betekenis van namen niet geleerd worden door *ostensie*. De betekenis van een zin is altijd relatief aan een situatie, aan een taalspel. Het is niet mogelijk het wezen van een taalspel, en daarmee het wezen van de taal, te beschrijven.

Ook Austin laat zien dat de betekenis van taal nauw verweven is met handelingen. Hij beschrijft taaldaden (*speech acts*), bijvoorbeeld het "Ja" bij de trouw-plechtigheid, waarmee geen bewering wordt gedaan maar een handeling wordt verricht.

Een andere bekende kritikus van de afbeeldingstheorie is de Amerikaan Quine. Hij benadrukt het holistische karakter van de relatie tussen onze (in taal uitgedrukte) overtuigingen en de ervaring. Kennisclaims hebben volgens hem geen zelfstandige empirische betekenis maar kunnen alleen worden begrepen als onderdeel van een theorie of *conceptual scheme*. Quine benadrukt dat dit des te sterker geldt naarmate de claims dieper zijn ingebed in de taal en dus moeilijker kunnen worden herzien.

> It is of such sentences above all that Wittgenstein's dictum holds true: "Understanding a sentence means understanding a language".
> – Quine [22, p. 73]

Quine gebruikt pragmatische doelstellingen: er zijn geen analytische waarheden, zelfs niet in de logika. Als in de kwantum-mechanika de wet van de uitgesloten derde moet worden verworpen, kan dat. Natuurlijk moet altijd eerst verworpen worden wat niet zo diep in het taalsysteem is geworteld.

Deze nieuwe kijk op taal is terug te vinden in vele moderne stromingen in de filosofie, bijvoorbeeld in de semiotiek, het strukturalisme, de hermeneutiek en de grammatologie. Deze stromingen zullen hier kort worden besproken.

De semiotiek is ontstaan uit een kritiek op het atomisme en benadrukt het verschil tussen *langue*, het systeem, en *parole*, het gesprokene. De taal wordt gezien als een systeem van tekens, een dynamische samenstelling van relaties. Taaluitspraken, de *discours*, moeten een strukturele analyse ondergaan om begrepen te worden. In het strukturalisme worden de strukturen van de *discours* statisch opgevat.

De hermeneutiek van Gadamer en de grammatologie van Derrida zijn geïnspireerd door het denken van Heidegger. In de hermeneutiek staat het overbrengen van betekenissen van een vreemde wereld naar onze bekende wereld centraal[4], bijvoorbeeld van de tekst naar de lezer. De hermeneutische overdracht veronderstelt een kunst; een vast, objektief of wetenschappelijk patroon van regels voldoet (weer) niet. De taal is het universele medium van de hermeneutische ervaring: "Welt haben heißt ... Sprache haben." [22, p. 145]

In de grammatologie wordt de taal niet meer gezien als een systeem van tekens die telkens zijn samengesteld uit een betekenaar en een betekende (als in de semiotiek) maar als een spoor, een reeks van betekenaren, die telkens weer naar nieuwe betekenden verwijzen.

[4] De term hermeneutiek voert terug naar de Griekse god Hermes die de boodschappen van de goden overbracht naar de wereld van de stervelingen.

2.2.3 Wetenschapsfilosofie

Misschien is de wetenschap op het moment nog eerder bekend om haar vermogen de mens van zijn vreugden te beroven en hem kouder, standbeeldachtiger, stoïcijnser te maken. Maar zij zou ook nog ontdekt kunnen worden als de grote pijnbrengster *– en dan zou wellicht meteen ook haar tegenkracht ontdekt zijn, haar ontzaglijk vermogen nieuwe sterrewerelden van vreugde te doen oplichten.*

– Friedrich Nietzsche [45, p. 49]

De wetenschap pretendeert kennis te leveren die dichter bij de waarheid ligt en die betrouwbaarder is dan wat met meer alledaagse middelen is vergaard of ons uit overlevering bekend is. Voor nieuwe wetenschappelijke resultaten ruilt men graag oude wijsheden in. In de wetenschapsfilosofie tracht men een karakterisering van de wetenschappen te geven en te onderzoeken in hoeverre de bovengenoemde aanspraken van wetenschap gerechtvaardigd zijn [62].

In de wetenschappen leert men op een andere wijze dan wij mensen op school leren. Dit wordt hoofdzakelijk uit pragmatische overwegingen gedaan. Bedenk maar eens hoeveel tijd het kost om een kind rekenen te leren. Jarenlang tafels uit het hoofd leren tot uiteindelijk enig rekenkundig gevoel is bijgebracht, voor ingewikkelder problemen werkt zo'n systeem niet. In de wetenschappen beheerst een onderzoeker meestal slechts een klein gebied, waarbinnen hij ekspert is. De kennis in een wetenschap is de som van de kennis van alle eksperts op dat gebied.

De theorieën over hoe men leert in de wetenschap, of zou moeten leren, zijn de afgelopen eeuw tweemaal ingrijpend gewijzigd. De eerste theorie is het *logisch positivisme*, waarin wordt gesteld dat wetenschappelijke theorieën op ervaring zijn gebaseerd en kennis groeit door accumulatie van 'feiten'. Induktie is de enige juiste methode van leren, uitgangspunten zijn de logika en de afbeeldingstheorie.

De belangrijkste reaktie op het *logisch positivisme* is de theorie van Popper, die stelde dat induktie overbodig is in de wetenschap. Groei van kennis is een gevolg van vrijmoedig verzonnen theorieën die aan 'kritische toetsen' kunnen worden onderworpen. Hoe een theorie tot stand komt is niet van belang.

De laatste theorie bestaat uit de kritiek van Kuhn op de eerste twee stromingen. Hij stelt dat er voor toetsing en methodologische kriteria nauwelijks enige rol is weggelegd in de feitelijke ontwikkeling van wetenschap.

Logisch positivisme

Rond 1930 ontstond in Wenen een invloedrijke stroming, de *Wiener Kreis*. Bekende aanhangers van deze stroming waren Carnap, Neurath en Schlick. Zij beoogden een omwenteling in de filosofie. Hun ideaal was de helderheid en konsistentie van de natuurwetenschappen. In plaats van informele spekulaties moest er een wetenschappelijk wereldbeeld ontstaan.

De *Wiener Kreis* wordt gekenmerkt door drie uitgangspunten. Het eerste is dat voor de wetenschappen een empiristische grondslag moet en kan worden gegeven. Dit betekent dat alle betrouwbare kennis op ervaring berust. Empirische wetenschap is opgebouwd uit een aantal uitspraken die Carnap *protokolzinnen* noemt. Een voorbeeld van een protokolzin is: "De ampèremeter wijst x mA aan". Deze uitspraken vormen de *rockbottom of knowledge*. De protokolzinnen komen overeen met de atomen van Wittgenstein in de afbeeldingstheorie.

Vervolgens namen de logisch positivisten aan dat alle wetenschappen dezelfde kennistheoretische struktuur bezitten: de *eenheid der wetenschappen*. Deze struktuur wordt onderzocht aan de hand van de meest ontwikkelde wetenschap, de natuurkunde.

De laatste aanname is dat een filosofische beschouwing aan de praktijk voorbij kan gaan. Dit betekent dat overwegingen die een onderzoeker tot een theorie hebben gebracht en de redenen die zijn collegae hebben om de theorie te accepteren of te verwerpen, de *context of discovery*, niet van belang zijn. Het gaat uitsluitend om de rechtvaardiging van de kennis zelf, de *context of justification*. Deze rechtvaardiging bestaat uit twee delen. In de eerste plaats moet worden onderzocht of de theorie wel thuis hoort in het wetenschappelijke domein. Indien dit het geval is kan de geldigheid van de theorie onderzocht worden aan de hand van empirisch feitenmateriaal.

De filosoof Wittgenstein leverde de eerste aanzet voor het gezochte kriterium voor kennis. In zijn beroemde *Tractatus logico-philosophicus* schreef hij:

> 4.024. Einen Satz verstehen, heißt, wissen was der Fall ist, wenn er wahr ist. [64, p. 44]

Een andere inspirator was Einstein[5], die het begrip tijd 'operationeel' definieerde in termen van gelijktijdigheid van gebeurtenissen. Carnap en Schlick kwamen toen op het beroemde verifikatiekriterium van betekenis: *de betekenis van een uitspraak is zijn methode van verifikatie.*

Carnap neemt later afstand van het rigide verifikatiekriterium en formuleert het *konfirmatiekriterium*: uitspraken moeten kunnen worden bevestigd door waarnemingen. Zo ontstaat de konfirmatiegraad van een theorie. Idealiter is de konfirmatiegraad één, het is dan zeker dat de theorie waar is. In de praktijk zal een theorie echter altijd een hypothetisch karakter hebben.

Carnap c.s. redeneert als volgt: een theorie is een verzameling uitspraken en kan worden geaxiomatiseerd. Dat wil zeggen dat er een aantal uitspraken kan worden gekozen, de axiomata, waarvan alle uitspraken die samen de theorie vormen met behulp van logische en wiskundige bewerkingen kunnen worden afgeleid (gededuceerd). De studie van een theorie kan zich dan beperken tot deze axiomata.

Een logisch systeem bestaat dus uit een aantal axiomata en een deduktietheorie waarmee feiten kunnen worden afgeleid uit deze axiomata. Als wordt

[5] Deze definitie wordt meestal aan Einstein toegeschreven maar is door Poincaré als eerste opgesteld.

verondersteld dat mensen op een logische manier denken en beweren – zoals in de afbeeldingstheorie – dan is de vraag hoe wij aan deze axiomata komen. Met andere woorden, de vraag is hoe wij onze kennis reduceren tot een aantal axiomata.

Kennis die wij uit waarnemingen verkrijgen is altijd eindig, terwijl een axiomatische theorie oneindig veel uitspraken kan afleiden. Een axiomatische theorie is dus algemener dan kennis verkregen uit observaties. De wijze van redeneren waarbij besloten wordt van het bijzondere tot het algemenere heet induktie.

De logisch positivisten proberen de axiomata te vinden door verifiëerbare regels op te stellen. Het systeem dat Carnap heeft trachten te formuleren wordt ook wel een induktieve logika genoemd.

In het dagelijks leven maken we veelvuldig gebruik van induktie. Als iedere dag (behalve in het weekend) de postbode langskomt om vier uur, gaan wij ervan uit dat hij morgen ook weer om vier uur komt (tenzij het morgen zaterdag of zondag is). En als wij een paar keer geluk hebben in het casino dan denken wij dat ons systeem winnend is (al weten we onbewust dat er geen winnend systeem kan bestaan).

In de wetenschappen is natuurkunde het meest sprekende voorbeeld van het gebruik van induktie. Zoals Newton reeds zei:

> We are to look upon propositions inferred by induction from phenomena.

Het probleem bij induktie is dat het geen logisch geldige redeneervorm is. Dit staat bekend als het induktieprobleem van Hume. Het betekent dat uit een (eindig) aantal uitspraken geen algemeen geldige wetten (over oneindig veel objekten) kunnen worden afgeleid. Neem bijvoorbeeld de waarneming van een willekeurig aantal zwarte kraaien, dan is niet bewezen (en kan ook niet worden bewezen) dat alle kraaien zwart zijn. Het is altijd mogelijk dat er een witte kraai bestaat, die de waarnemers echter nog niet gezien hebben.

Popper

Aan het einde van de 19^e eeuw pleitte Peirce, grondlegger van het pragmatisme, reeds voor een andere kijk op kennis, namelijk dat kennis geen fundament hoeft te hebben (als de *rockbottom of knowledge* van de logisch positivisten). Toch werd het gangbare beeld van wetenschap tot het begin van de jaren zestig door het logisch positivisme getekend.

Karl Popper breekt met de opvatting dat wetenschap op ervaringen berust, maar houdt wel vast aan de stelling dat er een empirische basis is voor wetenschap, een theorie heeft een empirische inhoud. Induktie speelt volgens hem echter geen rol in wetenschap, *ad hoc* aanpassingen zijn ook toegestaan. Hij stelt dat het geven van een fundering van wetenschappelijke kennis niet van belang is voor een filosoof; van belang is hoe kennis kan worden verbeterd. In 1934 verscheen reeds zijn beroemde boek *Logik der Forschung*, maar pas in 1959 bij de Engelse vertaling van het boek als *The logic of scientific discovery* brak hij door en werd zijn theorie als maatstaf gebruikt in de wetenschap.

Popper heeft twee argumenten voor zijn stelling dat induktie overbodig is in de wetenschap. Het eerste is het induktieprobleem van Hume (dat induktie geen logisch geldige redeneervorm is).

Het tweede argument is niet gebaseerd op logische gronden maar is kennistheoretisch van aard. Er kan niet over *zuivere ervaring* of *naakte feiten* gesproken worden. Er is geen onderscheid tussen (problematische) theoretische en (onproblematische) waarnemingsuitspraken: alle uitspraken zijn theoretisch (maar sommige zijn meer theoretisch dan andere). Waarnemingsuitspraken zijn altijd relatief aan een bepaald gezichtspunt en nooit een reportage van louter zintuiglijke indrukken.

Popper stelt dat wij leren van onze fouten. Objektieve kennis is kennis die onder kritiek gesteld kan worden. Er zijn vele bronnen van kennis en geen enkele heeft grotere zeggenschap dan een andere. Er is geen *rock-bottom of knowledge*: theorieën en hypothesen hebben de status van vermoeden en zullen die ook altijd houden. In de wetenschap dient naar waarheid te worden gezocht maar er dient niet naar zekerheid te worden gestreefd.

Popper formuleert een nieuw kriterium voor wetenschappelijke kennis, het *falsifikatiekriterium*: een theorie moet uit falsificeerbare uitspraken bestaan. Een uitspraak is falsificeerbaar als er een observatie mogelijk is waardoor de uitspraak wordt tegengesproken. Algemene uitspraken als "*p bestaat*" zijn niet te falsificeren, slechts te verifiëren. Als er een p gevonden wordt is de uitspraak geverifieerd, als er niet één gevonden wordt betekent dat nog niet dat er niet één bestaat. De uitspraak "*p bestaat niet*" is echter wel te falsificeren, als er zo'n p wordt gevonden is de uitspraak gefalsificeerd. Basisuitspraken als de protokolzinnen van het logisch positivisme zijn hypothesen van lagere orde, ook die kunnen worden gefalsificeerd. De uitspraken of observaties die een theorie kunnen falsificeren zijn de *falsifikatoren* van die theorie, een falsifikatie heet een *refutation*.

Popper stelt voor dat de wetenschappelijke activiteit gericht moet zijn op het opstellen van eenvoudige, te falsificeren theorieën (*conjectures*) die de bekende feiten verklaren en dat de eksperimenten erop gericht moeten zijn deze theorieën te falsificeren, dus op het vinden van een falsifikator. De interaktie tussen theoretische en eksperimentele aktiviteit, tussen de *conjectures* en *refutations*, zal in limiet konvergeren naar de waarheid. Dit is het antwoord van Popper op het induktieprobleem van Hume: de wetenschap is gebaseerd op het feit dat als een theorie fout is, zij ooit zal worden ontkend. Het is echter in het algemeen niet mogelijk te bewijzen dat een theorie juist is, dit is de "*conjectural status of scientific theories*".

Een theorie moet zoveel mogelijk uitsluiten. Zij bevat meer informatie naarmate zij meer uitsluit, naarmate zij meer falsifikatoren heeft. Dit kan worden gemeten met de *verisimilitude* van een theorie. De *verisimilitude* van een tautologie of een kontradiktie is nul, een theorie die precies alle ware feiten afleidt heeft een *verisimilitude* van één.

Het kriterium van Popper is geen zinkriterium als het verifikatiekriterium. Volgens de Wiener Kreis zijn niet-wetenschappelijke uitspraken niet zinvol, volgens Popper kunnen zij dat wel zijn. Neem bijvoorbeeld het meta-fysische idee van de atoomtheorie, volgens Popper was dit wel zinvol, volgens de Wiener Kreis niet. Het is Popper met het falsifikatiekriterium vooral te doen om de wetenschap af te scheiden van pseudo-wetenschappen als psychoanalyse, marxisme en astrologie. Wie namelijk een kennissysteem erop nahoudt dat niet falsificeerbaar is, heeft altijd gelijk. Het is niet mogelijk deze kennis te verbeteren, hetgeen bij falsificeerbare kennis wel mogelijk is.

Empirisch holisme

De belangrijkste kritiek op Popper is geformuleerd door T.S. Kuhn in *The structure of scientific revolutions*. Kuhn stelt:

- Er is geen sprake van groei van kennis in de richting van de waarheid.
- Er zijn geen algemene regels te formuleren die onderzoekers dienen te hanteren bij hun wetenschappelijke werk.
- Er zijn geen vaste methodologische regels aan de hand waarvan keuzen tussen theorieën kunnen worden gemaakt.
- Er is geen scherp onderscheid tussen kwesties die het ontstaan van kennis betreffen en vragen betreffende de beoordeling en rechtvaardiging van kennis.

Kuhn is in de eerste plaats wetenschapshistorikus en zijn kritiek ontstaat uit een analyse van de geschiedenis van de wetenschap. De wetenschap wordt gekenmerkt door twee fundamenteel verschillende typen perioden. Hij noemt deze typen *normale wetenschap* en *revolutie*.

Normale wetenschap wordt gekarakteriseerd door de overeenstemming die in zo'n periode te vinden is tussen wetenschappers. Men is het erover eens welke problemen het waard zijn te worden onderzocht en aan welke standaarden oplossingen moeten voldoen. Deze overeenstemming berust niet op gedeelde regels en definities, zoals Popper stelde, maar bestaat erin dat de onderzoekers in zo'n gemeenschap bepaalde prestaties van hun voorgangers als maatgevend beschouwen. Zo'n prestatie funktioneert als typisch voorbeeld, als *paradigma*, voor het soort aan te pakken problemen en de manier waarop deze problemen moeten worden aangepakt.

Tijdens een *revolutie* vervallen deze paradigmata. Als geaccepteerd is dat het tot nu toe gehanteerde wetenschappelijke kader niet toereikend is, vallen ook de standaarden voor het wetenschappelijk onderzoek weg. De uitkomst van een revolutie, het nieuwe kader, kan niet met onafhankelijke kriteria worden beoordeeld; er is geen rationele verklaring voor de belangrijkste overgangen in de wetenschap.

Lakatos bekritiseert Kuhn en noemt hem een *elitist*, omdat zijn ideeën over wetenschap uitgaan van een soort gewoonterecht waarbij de enige rechter over wetenschap de onderzoekers zelf zijn. De waarheidsvraag is door Kuhn getransformeerd tot een machtsvraag.

Lakatos werkt zijn kritiek uit in een methodologie van onderzoeksprogramma's. Wetenschappelijk werk dat door een filosoof onder de loupe wordt genomen is niet een theorie, zoals bij Popper, maar een keten van theorieën. Lakatos stelt een aantal kriteria op waaraan de ontwikkeling van wetenschap binnen dergelijke programma's moet voldoen. Bij het opstellen van deze kriteria is hij sterk beïnvloed door Popper. Met behulp van deze kriteria kunnen ook *revoluties* op een rationele manier worden beslecht.

2.3 Kunstmatige intelligentie

Het geven van een goede definitie van AI is moeilijk omdat het begrip intelligentie binnen de filosofie nog niet erg duidelijk is. Bonnet geeft de volgende definitie:

> AI is de tak van wetenschap die zich ten doel stelt een beter inzicht te verkrijgen in de aard van intelligentie met behulp van komputerprogramma's waarmee menselijke intelligentie wordt gesimuleerd. [7, p. 3]

Sommige onderzoekers nemen de definitie nog ruimer:

> But many others would favour a more controversial definition, seeing AI as *the science of intelligence in general* – or, more accurately, as the intellectual core of cognitive science. [5]

Er kunnen twee stromen binnen de AI worden onderscheiden, *GOFAI* (Good Old Fashioned AI) en *connectionisme*. Beide zijn terug te leiden naar een artikel van McCulloch en Pitts uit 1965, *Logical calculus of nervous activity* [5]. Hierin bewezen zij dat bepaalde soorten neurale netwerken in principe bepaalde logische funkties kunnen berekenen. In de GOFAI ziet men de komputer als een systeem voor het manipuleren van mentale symbolen, in het connectionisme als een manier om het brein te modelleren [5, p. 309].

2.3.1 GOFAI

In GOFAI is het oplossen van problemen het paradigma van intelligentie. Dit gebeurt door het manipuleren van symbolen. De theoretische fundering van GOFAI werd gelegd door Alan Turing, die berekeningen definieerde als de formele representatie van (niet geïnterpreteerde) symbolen door het toepassen van formele regels. Newell en Simon stelden aan de hand van dit uitgangspunt twee hypothesen op:

> *The Physical-Symbol System Hypothesis.* A physical-symbol system has the necessary and sufficient means for general intelligent action. [5, p. 111]

> *The Heuristic Search Hypothesis.* The solutions to problems are represented as symbol structures. A physical-symbol system exercises its intelligence in problem-solving by search – that is, by generating and progressively modifying symbol structures until it produces a solution structure. [5, p. 119]

Voor de representatie van kennis hebben deze twee hypothesen als konsekwentie dat symbool-strukturen als basiselementen kunnen worden gerepresenteerd, wat door Brian Smith de kennisrepresentatie-hypothese is genoemd:

> Ieder intelligent proces bevat strukturele ingredienten die bepaalde kennis van het proces representeren. [50, p. 1]

In het klassieke *Computing, machinery and intelligence* [5, p. 40] uit 1950 vraagt Turing zich af of machines die symbolen kunnen manipuleren intelligent zijn, of zij kunnen denken. Hij was van mening dat dit inderdaad het geval is en definieerde een intelligentie-test voor machines, het imitatiespel, waarbij de machine intelligent genoemd wordt als hij een menselijk gesprek kan imiteren. Hierop kwamen veel protesten, waaronder de bekendste van Searle in *Minds, brains and programs* [5, p. 67].

GOFAI is een opvolger van de reduktionistische taaltheorieën van Russell en Wittgenstein, de symbolen zijn de atomen van de afbeeldingstheorie. Als representatieformalisme gebruikte men eerst de logika en later andere formalismen, maar men hield vast aan bovenstaande hypothesen.[6]

Kennis wordt in een GOFAI-systeem opgeslagen in de vorm van symbolen en afleidingsregels. De totale kennis van het systeem is dan gelijk aan alle uitspraken die kunnen worden afgeleid. Als deze kennis moet worden uitgebreid, als het systeem ook moet kunnen leren, dan zullen daar aparte mechanismen voor moeten worden geschreven. Deze mechanismen zijn meestal een vorm van induktie. Zo is GOFAI dus ook een opvolger van het logisch positivisme.

De eerste successen gaven de onderzoekers hoop op een snelle en spektakulaire opmars van intelligente machines. In 1958 schreef Simon:

> It is not my aim to surprise or shock you ... But the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until – in a visible future – the range of problems they can handle will be coextensive with the range to which the human mind has been applied. [5, p. 312]

De geschiedenis echter heeft deze hooggespannen verwachtingen niet beantwoord. Het ideaal van een machine die evenveel kan als wij mensen kunnen zal voorlopig niet worden bereikt. Daarom werden de doelstellingen gewijzigd.

[6] Zie [49] voor een uitgebreide behandeling en diskussie van taaltheorie in relatie met AI.

In 1978 schreef Patrick Hayes het *Naïve physics manifesto* [5, p. 171], een oproep om een kennissysteem te bouwen waar common-sense kennis in kan worden opgeslagen.

Ook dit bleek te hoog gegrepen. McDermott schrijft in 1985 in *A critique of pure reason*:

> The logicist project of expressing 'naïve physics' in first-order logic has not been very successful. One reason may be that the basic argument was flawed. You cannot write down axioms independent of a program for manipulating them if inferences you are interested in are not deductions. Unfortunately, very few interesting inferences are deductions, and the attempts by logicists to extend logic to cover more territory have been disappointing. Hence we must resign ourselves to writing programs, and viewing knowledge representations as entities to be manipulated by programs. [5, p. 227]

2.3.2 Connectionisme

In het connectionisme probeert men de neurale struktuur van de hersenen te kopiëren. Niet 'het oplossen van problemen' maar 'het leren' geldt als paradigma van intelligentie. Er wordt geen gebruik gemaakt van logika maar van statistiek.

Kennis binnen een connectionistisch systeem bestaat uit een relatie tussen in- en uitvoer. Kennis wordt niet expliciet in het systeem ingevoerd, zoals bijvoorbeeld de axiomata in een logika. In plaats daarvan worden een aantal voorbeelden tijdens een leerproces in het systeem ingevoerd. Deze voorbeelden gelden als paradigmata van de kennis.

Een voorbeeld van het leerproces is het leren van werkwoorden. Het probleem bij het leren van werkwoorden zijn de onregelmatige werkwoorden. Een GOFAI systeem zal de regelmatige werkwoorden axiomatiseren ("hij is stam plus t") en worden uitgebreid met een lijst van uitzonderingen. Een connectionistisch systeem krijgt tijdens het leerproces een groot aantal werkwoorden als invoer en de derde persoon enkelvoud als gewenste uitvoer. Na het leerproces kan dan een willekeurig werkwoord als invoer worden gegeven en de uitvoer is dan (hopelijk) de juiste derde persoon enkelvoud.

De belangrijkste stroming in het connectionisme is de *parallel distributed processing* (PDP) van Hinton, McClelland en Rumelhart.[7] Het verschil van kennisrepresentatie blijkt uit het volgende citaat:

> Information is not stored anywhere in particular. Rather, it is stored everywhere. Information is better thought of as 'evoked' than 'found'.
> – Rumelhart en Norman [5, p. 309]

Het connectionisme kan worden gezien als opvolger van de holistische taaltheorieën. De kennis is niet te axiomatiseren, de uitvoer is afhankelijk van het

[7] PDP wordt soms vereenzelvigd met connectionisme [50], connectionisme is echter een ruimer begrip.

hele connectionistische systeem.

Een vraag die blijft is welke voorbeelden tijdens het leerproces moeten worden ingevoerd om het systeem te laten leren. De kennis van het systeem is afhankelijk van deze voorbeelden (en de volgorde waarin zij ingevoerd worden). Er zijn geen regels waaraan de voorbeelden moet voldoen in de zin van verifikatie of falsifikatie, sommige van de voorbeelden mogen zelfs fout zijn! Een connectionistisch systeem voldoet dus aan de kritiek van Kuhn op Popper en de logisch positivisten.

Sommige AI-problemen zijn goed te formaliseren met behulp van een GOFAI-theorie, bijvoorbeeld wiskundige problemen als de Fourier-transformatie. Als er voor een probleem op het gebied van informatie-verwerking een algoritme kan worden gevonden, noemt David Marr zo'n probleem een Type-1 theorie. Maar niet voor alle problemen hoeft er een algoritmische oplossing te bestaan:

> The fly in the ointment is that while many problems of biological information-processing have a Type-1 theory, there is no reason why they should all have. This can happen when a problem is solved by the simultaneous action of a considerable number of processes, *whose interaction is its own simplest description*, and I shall refer to such a situation as a Type-2 theory. – David C. Marr [5, p. 134]

Het onderzoek in de AI is sterk gedomineerd door GOFAI. Minsky gaf een vernietigend oordeel over perceptrons, een representatieformalisme binnen het connectionisme:

> Perceptrons have been widely publicized as 'pattern recognition' or 'learning' machines and as such have been discussed in a large number of books, journal articles and voluminous 'reports'. Most of this writing ... is without scientific value. – Marvin Minsky [5, p. 315]

Het gevolg was dat het merendeel der AI-onderzoekers niets meer te maken wilde hebben met connectionisme. In de tachtiger jaren echter werden de verwachtingen van GOFAI niet beantwoord en werden met connectionistische systemen enige indrukwekkende resultaten behaald. Zo werd een systeem geschreven voor het uitspreken van Engelse woorden, iets waar men binnen GOFAI nog niet in was geslaagd. Het connectionisme kende het afgelopen decennium een opleving; GOFAI en connectionisme staan nu gelijkwaardig naast elkaar. In de toekomst zal moeten blijken welke stroming het beste de menselijke intelligentie kan simuleren.

2.3.3 Kennisrepresentatie

Bij kennisrepresentatie gaat het niet meer om de vraag welke kennis tot het domein behoort of hoe kennis gerechtvaardigd is, maar hoe een informatie-systeem het beste in een komputer kan worden geïmplementeerd. In kennisrepresentatietalen wordt informatie gerepresenteerd zonder te eisen dat deze informatie waar of gerechtvaardigd is en het is dan ook beter te spreken over *informatierepresentatie* dan over kennisrepresentatie [50, p. 3].

Een GOFAI-kennissysteem bestaat uit twee delen, de basiskennis en een afleidingsgedeelte. De basiskennis geeft aan welke kennis expliciet is opgenomen, het afleidingsgedeelte geeft aan welke kennis impliciet is en via afleidingsmechanismen uit de basiskennis kan worden afgeleid.

Een belangrijke diskussie met betrekking tot kennisrepresentatieformalismen gaat over de tegenstelling tussen procedurele en deklaratieve representaties. Dit is de tegenhanger van de filosofische diskussie over de tegenstelling tussen *weten hoe* en *weten dat* [50].

Bij een procedurele representatie wordt ervan uitgegaan dat kennis hoofdzakelijk *weten hoe* is, dus bestaat uit heuristieken (ervaringsregels). Kennis kan worden gerepresenteerd door een verzameling procedures te schrijven die ieder een stuk kennis representeren.

Bij een deklaratieve representatie wordt er vanuit gegaan dat kennis hoofdzakelijk *weten dat* is, dus feiten en overtuigingen. Kennis bestaat uit een aantal symbolen die deze feiten en overtuigingen representeren, in plaats van deze feiten in procedures te 'verstoppen'. Afleidingen worden identiek gezien en worden maar één keer gedefinieerd. Kowalski definieerde deze identiteit als: algoritme = logika + besturing [50].

Opslag van kennis is in een deklaratief programma veel efficiënter en leeralgoritmen kunnen eenvoudiger worden geïmplementeerd. Voor een procedurele aanpak pleit dat veel kennis die wij bezitten procedureel is. Een voorbeeld hiervan is de manier waarop wij rekenen. Dat zou kunnen met de axiomata van Peano [17], maar dat is zeer tijdrovend. Procedures zijn vaak efficiënter qua rekentijd.

Een ander belangrijk aspekt van kennissystemen is de kontrole over de afleidingen die kunnen worden gemaakt. Er zijn twee vormen van redeneren: *forward* en *backward reasoning*. Bij *forward reasoning* wordt uitgegaan van de kennis in het systeem en wordt getracht een bepaalde uitspraak af te leiden. Bij *backward reasoning* wordt uitgegaan van de te bewijzen uitspraak en wordt getracht te bewijzen dat hij in het systeem past.

Er zijn twee manieren om de efficiëntie van het zoekproces te verbeteren. Ten eerste kunnen er soms heuristieken worden gebruikt die bepalen of een afleiding nog wel zinvol is. Een andere manier om de efficiëntie te verbeteren is het toevoegen van expliciete besturingskennis. Een deklaratieve taal met expliciete besturingskennis komt overeen met een procedurele taal, het voordeel ten opzichte van een procedurele taal is dat er onderscheid wordt gemaakt tussen verschillende soorten kennis (domeinkennis en strategische kennis) en dat systemen gemakkelijker te ontwikkelen en onderhouden zijn.

Globaal gezien zijn er drie stromingen binnen de kennisrepresentatie, twee binnen de GOFAI en één binnen het connectionisme, met ieder een andere eenheid van kennis.

De formalismen van de eerste stroming zijn de logika's en produktiesystemen. In deze formalismen worden de basisuitspraken, de atomen of symbolen, zo klein mogelijk gekozen.

De formalismen van de tweede stroming zijn semantische netten en frames, waarin de primitieve stukken kennis veel groter zijn dan in de formalismen van de eerste stroming. Hiermee kan in het formalisme de verbondenheid van bepaalde gegevens worden benadrukt door ze bij elkaar te groeperen binnen een objekt.

De derde stroming is de connectionistische representatie, de neurale netwerken. In een neuraal netwerk is het gehele netwerk de kleinste eenheid van kennis.

Logika en produktiesystemen

De logika is het bekendste formalisme voor kennisrepresentatie. Binnen de logika kan een onderscheid worden gemaakt tussen monotone en niet-monotone logika's. Een monotone logika is een logika waarin een konklusie die uit een bepaalde hoeveelheid kennis kan worden getrokken, ook nog getrokken kan worden als de kennis toeneemt. Dit komt overeen met de redeneerprocessen in de werkelijkheid, waarbij een onvoorwaardelijke konklusie wordt getrokken. In een niet-monotone logika kunnen ook *ceteris paribus* konklusies worden getrokken. De klassieke logika's zijn voorbeelden van monotone logika's, niet-monotone logika's zijn pas gedurende de laatste paar decennia ontworpen.

Het voordeel van een monotone logika boven een niet-monotone is dat afleidingen veel efficiënter kunnen worden gevonden. Nadelen van een monotone logika zijn dat overtuigingen niet gerepresenteerd kunnen worden (alleen feiten) en dat *default reasoning* en *abduction* niet kunnen worden geformaliseerd [5, p. 206].

Default reasoning is een vorm van redeneren waarbij plausibele gevolgtrekkingen worden gemaakt op basis van afwezigheid van strijdige informatie [50, p. 62].

Mensen hanteren vaak *default reasoning* in common-sense redeneringen. Een voorbeeld is de uitspraak dat ik van de zomer op vakantie ga. Ik ga ieder jaar op vakantie, dus zal ik dat dit jaar waarschijnlijk weer doen. Maar zeker is het niet, ik zou bijvoorbeeld ziek kunnen worden[8]. Zolang ik echter nog niet ziek ben (of er iets anders gebeurt), zolang er geen strijdige informatie is, ga ik ervan uit de komende zomer enige weken vakantie te houden (de *default* gebeurtenis).

Abduction[9] is het vinden van de meest waarschijnlijke verklaring. Beschouw als voorbeeld de logische konsekwentie “als het heeft geregend, zijn de straten nat.” Stel nu dat de straten nat zijn, dan mag ik nog niet konkluderen dat het heeft geregend (misschien is er een waterleiding gesprongen). Deze gevolgtrekking is abduktief: de hypothese dat het heeft geregend is de meest voor de hand liggende uitleg.

Er zijn verschillende niet-monotone logika's, waarvan de bekendste *circumscription* van McCarthy en *default logic* van Raymond Reiter zijn. Hier wordt een voorbeeld gegeven van een niet-monotone logika waarin de *closed world assumption* (CWA) geldig is. Een logika bevat vaak onbepaalde uitspraken, uitspraken

[8] Of deze skriptie kost meer tijd dan gepland ...

[9] Een term geïntroduceerd door de filosoof Peirce.

waarvan niet bekend is of zij waar of onwaar zijn. De CWA veronderstelt dat alle uitspraken die onbepaald zijn ook onwaar zijn.

Wat betreft mijn vakantieplannen voor deze zomer is er in die logika een regel: *Als ik niet ziek ben en als mijn skriptie niet vertraagd is dan ga ik van de zomer op vakantie.* Het is niet bekend of ik ziek zal zijn, dus wordt middels de CWA aangenomen dat ik niet ziek zal zijn en dat ik dus op vakantie ga. Word ik nu toch ziek dan wordt dit feit aan de kennis toegevoegd en kan via deze regel niet meer worden afgeleid dat ik op vakantie ga, dus ga ik niet op vakantie.

Het is hierbij van belang welke interpretatie de symbolen hebben, want als een onbepaald symbool staat voor *ik ben niet ziek* dan wordt aangenomen dat ik wel ziek ben.

Het principe van de CWA wordt ook wel het struisvogel-principe genoemd: als er onzekerheid bestaat over een uitspraak wordt aangenomen dat de uitspraak waarschijnlijk onwaar is, wat onbekend is kan ons niet schaden.

Overtuigingen kunnen wel in een niet-monotone logika of in een *reason maintenance system* (RMS) worden geïmplementeerd. Een RMS is een uitbreiding van een kennissysteem dat is gebaseerd op pure logika [50, p. 77] en is een methode om te voorkomen dat er tegenspraken in een kennissysteem kunnen worden afgeleid. In de RMS van Doyle [50, p. 75] krijgen alle uitspraken die afgeleid worden een *support status* en een *rechtvaardiging*. De *rechtvaardiging* geeft weer via welke andere uitspraken de uitspraak is afgeleid, de *support status* geeft aan of een uitspraak nog geldig is binnen het systeem. Als er een tegenspraak wordt afgeleid binnen het kennissysteem, dus als een bepaalde uitspraak waar is en zijn ontkenning ook, dan wordt van één van beide uitspraken de *support status* op ongeldig gezet. Bovendien worden ook de veroorzakers en de konsekwenties ongeldig verklaard, zodat het hele systeem geen tegenspraken meer af kan leiden.

In het vakantievoorbeeld wordt de overtuiging dat ik op vakantie ga afgeleid uit de overtuiging dat ik niet ziek zal zijn. Word ik toch ziek dan wordt de *support status* van de uitspraak "ik word niet ziek" ongeldig, via de rechtvaardigingen wordt gevonden dat de uitspraak "ik ga op vakantie" ervan afgeleid is en zijn *support status* wordt ook op ongeldig gezet.

Het principe van een RMS kan ook worden toegepast in lerende systemen. Als er een tegenspraak optreedt blijkt via de rechtvaardigingen welke geleerde regels niet korrekt zijn. Deze regels kunnen dan weer ongeldig worden gemaakt door de *support status* op ongeldig te zetten. Een dergelijk idee speelt ook bij het `Model Inference System` (`MIS`) van Shapiro [56]. Verder komt een RMS ook van pas bij niet-monotone logika's. Een nadeel van niet-monotone logika's is dat alle oude afleidingen waardeloos zijn geworden als de kennis toeneemt omdat ze misschien niet meer afleidbaar zijn. In een RMS kan via de rechtvaardigingen worden gekontroleerd welke uitspraken niet meer afleidbaar zijn. Dit kan efficiënter zijn dan alle afleidingen opnieuw maken.

Produktieregels lijken op afleidingen binnen een logika. Zij worden vaak toegepast in kennissystemen, maar ook bij simulatie van de manier waarop mensen problemen oplossen [50]. Een produktiesysteem is vooral geschikt voor patroon-gestuurde afleidingen [50, p. 80]. Zo'n afleiding mag worden gemaakt als een bepaald patroon voorkomt binnen de beschikbare kennis. Een produktiesysteem bestaat uit drie delen: het *werkgeheugen*, de *patroon-gestuurde modules* en een *interpreter*. Het *werkgeheugen* bestaat uit een aantal werkelementen waarin feitenkennis van het probleemgebied is opgeslagen. Een *patroon-gestuurde module* bestaat uit een patroon en een instruktie en representeert kennis over afleidingen. Als het patroon voorkomt in het *werkgeheugen* dan kan de instruktie worden uitgevoerd en deze instruktie is in staat de kennis van het werkgeheugen te veranderen. De *interpreter* heeft de kontrole over de afleidingen van de modules.

Overtuigingen kunnen in produktiesystemen worden opgenomen door de modules een waarschijnlijkheidsfaktor te geven. Een voorbeeld hiervan is de diagnose van ziekten: als een patient hoofdpijn heeft, dan is de kans dat dit door stress wordt veroorzaakt 0.79.

Een voordeel van een produktiesysteem boven een logika is de modulaire opbouw ervan. Een nadeel is dat het leren van kennis in een produktiesysteem moeilijk is omdat altijd de werking van de *interpreter* van belang is [50].

Semantische netten en frames

Logika's en produktiesystemen hebben twee samenhangende nadelen. Het eerste is dat individuele stukken kennis los van elkaar staan, terwijl er in de werkelijkheid een veel grotere verbondenheid bestaat tussen bepaalde kennis. Zo schreef Aristoteles reeds dat er in het menselijke geheugen een groot aantal verbindingen of associaties tussen verschillende stukken informatie aanwezig is [50].

Een tweede nadeel is het feit dat de eenheid van kennis in een logika of produktiesysteem erg klein is terwijl menselijke kennis lijkt te zijn opgeslagen in grotere stukken als een verzameling van entiteiten met bijbehorende beschrijvingen [50, p. 143].

Middels een implementatie kunnen de genoemde nadelen in een logika of produktiesysteem worden opgeheven, maar het wordt niet opgelegd in de taal en voor de gebruiker is de verbondenheid in een implementatie niet meer zichtbaar.

Als oplossingen voor deze problemen zijn de semantische netten en frames ontwikkeld. In een semantisch net ligt de nadruk op de verbondenheid tussen de kennis, in een frame ligt de nadruk meer op grotere eenheden kennis.

Semantische netten zijn in 1966 geïntroduceerd door Quillian [50] en worden hoofdzakelijk toegepast bij psychologisch onderzoek. Een semantisch net bestaat uit knopen en kanten tussen twee knopen. Knopen refereren naar objekten, of klassen van objekten, in de wereld. Een kant geeft een relatie tussen twee objekten aan. De kennis van de objekten bevindt zich op de kanten. Afleidingen komen neer op het volgen van kanten tussen knopen.

Semantische netten zijn in populariteit voorbij gestreefd door frames. Ze zijn in 1975 geïntroduceerd door Marvin Minsky, die een frame definieerde als een datastruktuur om stereotype situaties te representeren [50, p. 143]. In een frame wordt kennis over een bepaald aspekt van de wereld opgeslagen. De verschillende frames in een kennissysteem zijn verbonden via gerichte kanten die weergeven of een frame een deelverzameling is van een ander frame.

De verzameling van Rotterdammers is bijvoorbeeld een deelverzameling van de klasse Nederlanders die op zijn beurt weer een deelverzameling van de klasse mensen is. De afleidingen in een frame lopen via deze verbindingen en wordt *overerving* genoemd. De eigenschappen van een superklasse worden geërfd door een element, als iets voor alle mensen geldt, geldt dit ook voor de Rotterdammers. Dit idee komt ook terug in het objekt-georiënteerde programmeren (OOP), een programmeer-paradigma dat tegenwoordig erg populair is.

Neurale netwerken

In het connectionisme gaat men nog een stap verder dan in semantische netwerken en frames; er wordt gesteld dat slechts het kennissysteem als geheel als eenheid mag worden gezien. Kennis wordt opgeslagen in een neuraal netwerk, een kombinatie van een (groot) aantal knopen en kanten. De knopen en kanten refereren niet naar objekten of eigenschappen van objekten in de wereld maar hebben een aantal statistische eigenschappen.

Een kant heeft een waarde, de verbindingssterkte en een knoop heeft een drempelwaarde, de aktiveringswaarde. Een knoop krijgt invoerwaarden en geeft een uitvoerwaarde door aan de knopen waarmee hij verbonden is. Deze uitvoerwaarde wordt berekend aan de hand van de, middels de verbindingssterkten, gewogen som van de invoerwaarden. Het leren van een neuraal netwerk bestaat uit het veranderen van verbindingssterkten van kanten.

Een implementatie van een Type-1 theorie in een neuraal netwerk is veel minder efficiënt dan in een GOFAI-formalisme. Voor Type-2 theorieën zou een neuraal netwerk echter wel een betere oplossing kunnen zijn. In GOFAI tracht men reeds de verschillende formalismen te kombineren. De volgende stap is GOFAI formalismen met neurale netwerken te kombineren.

Een voordeel van neurale netwerken is dat zij eenvoudig te parallelliseren zijn. Als de techniek van parallellisatie verbetert, worden implementaties van neurale netwerken efficiënter.

Een belangrijke motivatie voor het gebruik van neurale netwerken is neurobiologisch. De menselijke intelligentie is 'geïmplementeerd' in de hersenen. Een neuraal netwerk heeft dezelfde struktuur als de hersenen. De hersenen bestaan uit een groot aantal zenuwcellen, neuronen genaamd, die onderling verbonden zijn met synapses. Neuronen kommuniceren met elkaar door een elektrische puls door een synaps te sturen. Deze impuls doet een chemische substantie vrijkomen in de ontvangende neuron. Daardoor wordt de status van dit neuron gewijzigd, mogelijkerwijs in die mate dat hij op zijn beurt ook weer elektrische pulsen gaat versturen ('omdat zijn drempelwaarde wordt overschreden').

De neurologie is nog niet erg ver gevorderd in het begrijpen van de werking van onze hersenen. Wellicht zullen de ontwikkelingen binnen de neurologie invloed hebben op de kunstmatige intelligentie:

> One need not endorse this suggestion (that AI models the whole brain) in arguing that AI should pay more attention to the brain than it has done so far. Despite its root in ideas about 'nervous activity', current connectionism owes very little to neuroscience. (Traditional AI owes even less, regarding questions about neural implementations as not only distinct from but utterly irrelevant to computational questions). – Margaret Boden [5, p. 19]

2.3.4 Induktie in logika

De logische basis voor automatische deduktie werd door Herbrand gelegd in de vroege jaren dertig en in 1965 werd een doorbraak bereikt door Robinson met zijn formulering van de afleidingsregel resolutie als basis voor mechanische deduktie. Berekeningen in de programmeertaal Prolog zijn gebaseerd op logische bewijzen door het toepassen van de resolutie-regel en Prolog was de programmeertaal van het Japanse vijfde-generatie-projekt[10].

Het probleem van een kennissysteem met logika als kennisrepresentatieformalisme is de keuze van de axiomata. Een intelligente machine zou deze axiomata zelf moeten kunnen ontdekken. Turing stelde reeds dat intelligente machines ook moeten kunnen leren [5, p. 40]. De meest voor de hand liggende methode voor leren binnen de logika is induktie.

In AI is induktie met succes toegepast. Er zijn kennissystemen met onvolledige kennis gebouwd die de ontbrekende kennis zelf aanvullen. Bij het bouwen van kennissystemen levert induktie een grote besparing op in bouwtijd en in onderhoud.

De induktieve modellen, voorgesteld in de literatuur, worden gemotiveerd door theorieën uit de wetenschapsfilosofie.

Drie modellen zullen hier worden besproken. Plotkin stelt een model op, gebaseerd op de filosofie van de Wiener Kreis. Muggleton en Buntine breiden dit model uit. Shapiro stelt een model op, gebaseerd op de filosofie van Popper.

Plotkin

Plotkin [48] stelt een formeel model op van algemene induktieve generalisatie en probeert vervolgens algoritmen te vinden voor de oplossing van dit model.

Hij erkent dat er geen onfeilbare methode is om generalisaties te vinden, het induktieprobleem van Hume, en noemt de generalisaties daarom hypothesen. Deze hypothesen hebben de vorm *als voor alle ... geldt dat ..., dan geldt voor hen ook ...*. Een hypothese geeft een zo goed mogelijke verklaring van verschijnselen door een relatie te leggen tussen data en bewijs van deze data.

[10] Onlangs bleek dat dit projekt de hooggespannen verwachtingen niet heeft beantwoord.

De generalisaties die het formele model levert kunnen door een robot gemaakt worden. Plotkin stelt zijn theorie echter op aan de hand van de wetenschapsfilosofie.

Dit motiveert hij door te stellen dat er een parallel is tussen de problemen van een wetenschapper met zijn geavanceerde kijk op de wereld en de robot met zijn primitieve kijk. Ook een robot zou in staat moeten kunnen zijn te zien en te handelen, hij heeft een oer-ontologie. De robot moet generalisaties kunnen maken om kausale verbanden tussen objekten en akties te kunnen leggen [48, p. 103].

Het formalisme van Plotkin is de clause logika, een deelverzameling van de eerste-orde predikatenlogika, maar met hetzelfde uitdrukkingsvermogen. Plotkin is optimistisch over de mogelijkheid dat alle concepten kunnen worden gevormd in de eerste-orde predikatenlogika:

> Our second assumption is that all concepts are to be formed in the first order predicate calculus. This is done, primarily, for technical convenience. It is accepted that the theories of science are amenable to the axiomatic method. Carnap (1976) is an impressive exponent of this view. ...We do not expect knowledge logics to enter the picture since the hypotheses refer to the world rather than the robot's relation to the world [48, p. 103].

Het model van Plotkin komt op het volgende neer: gegeven zijn een theorie en een aantal uitspraken die door de theorie niet kunnen worden afgeleid. De theorie noet zodanig worden uitgebreid dat ook deze nieuwe feiten kunnen worden afgeleid maar het systeem niet triviaal wordt (niet alles kan afleiden).

De theorie wordt niet aangepast, slechts uitgebreid. Het gaat hier dus om feiten, die altijd waar zijn en niet om overtuigingen, die later onwaar kunnen blijken.

Nieuwe concepten kunnen niet door het induktiemechanisme worden geïntroduceerd, het leren van een robot is beperkt tot concepten die hij reeds kent. In de termen van Kuhn is het leren beperkt tot *normale wetenschap*.

Een probleem met het model van Plotkin is dat het onbeslisbaar is, dat betekent dat er geen algoritme bestaat dat altijd de oplossing voor het model vindt. Dit is een gevolg van de onbeslisbaarheid van de eerste-orde predikatenlogika.

Plotkin heeft zijn model ook geïmplementeerd, maar het blijkt dat zijn implementatie niet erg efficiënt is.

Plotkin en Popper

Volgens Popper is induktie overbodig in de wetenschap; hoe een nieuwe theorie tot stand komt is niet doorslaggevend. Dit betekent niet dat de nieuwe theorie niet met behulp van induktie mag worden opgesteld. De enige eis is dat de theorie falsificeerbaar is.

Plotkins *logic of discovery* voldoet aan deze eis, want de hypothesen die worden opgesteld hebben alle de vorm *voor alle x geldt ...*, een x waar dit niet voor geldt is dus een falsifikator. Plotkin schrijft hierover:

> For Popperians, there can be no absolute justification of a hypothesis. There are instead a series of tests and competitions with other hypotheses. Their analysis is most concerned with universal laws, the form of scientific ones, for example: a law must be refutable and make many testable predictions. It is by making contradictory predictions that one of a competing pair may be eliminated. Thus ... it is suggested that simple hypotheses with a low *a priori* logical probability be sought. [48, p. 105]

Met behulp van het model van Plotkin kunnen verschillende theorieën worden opgesteld, waarvan sommige zullen worden gefalsificeerd. De theorie die het meest uitsluit (die met de hoogste *verisimilitude*) is de theorie die als standaard geldt.

Een praktisch probleem is dat er veel verschillende theorieën naast elkaar moeten bestaan, zowel zeer algemene (die wellicht snel worden gefalsificeerd) als theorieën met een zeer lage *verisimilitude*. Plotkins model is daarom geen efficiënte implementatie van de theorie van Popper.

Muggleton en Buntine

Muggleton en Buntine [31] breiden het induktiemechanisme van Plotkin uit. Er kunnen meer verschillende hypothesen worden verzonnen door het systeem, dus de juiste hypothese zal sneller (met behulp van minder voorbeelden) gevonden zijn. Om van de mogelijke hypothesen er één te kunnen selekteren wordt er gebruik gemaakt van een orakel: de algoritme genereert hypothesen en de gebruiker moet aangeven of een hypothese korrekt is of niet.

De belangrijkste uitbreiding is dat de taal uitgebreid kan worden, de zogenaamde *predicate invention*. Het leren beperkt zich niet tot gegeven concepten, de robot kan nieuwe concepten *uitvinden*.

Zij implementeerden een induktief systeem genaamd `CIGOL` (*logic* achterstevoren). `CIGOL` is efficiënter dan het systeem van Plotkin omdat er minder voorbeelden nodig zijn om de gezochte hypothese te vinden.

Als formalisme gebruikten zij een logika met een kleiner uitdrukkingsvermogen dan de eerste-orde predikatenlogika, de Horn-clause logika. Een voordeel van de Horn-clause logika is dat het efficiënter kan worden geïmplementeerd.

Het induktiemechanisme van Muggleton en Buntine is nog verder uitgebreid, onder andere door Matwin en Lapointe [29]. Wederom levert dit een grote winst in efficiëntie op (voor een aantal problemen).

Shapiro

Ook Shapiro [56] heeft een model opgesteld aan de hand van een theorie uit de wetenschapsfilosofie. Net als Plotkin ziet hij een overeenkomst tussen lerende systemen en wetenschappers:

> This paper is concerned with the folowing type of problem:
>
> > *Suppose we can acquire factual knowledge about a certain domain, which is governed by some unknown rules, or theory, and assume that a language adequate for expressing these rules is given. How can we discover these rules or theory from this factual knowledge?*
>
> A problem like this is faced by a scientist that is working in some domain under a fixed conceptual framework, performing experiments and trying to find true of the domain and capable of explaining their results. [56, p. 2]

Shapiro volgt de filosofie van Popper. Een *conjecture* is een verzameling van hypothesen, die worden gevonden door een kontinu proces van wijzigingen van de *conjecture* en *refutations* van de *conjecture*. Het zoeken naar hypothesen verloopt in Shapiro's model van algemene naar specifieke. Eerst wordt een zo algemeen mogelijke hypothese toegevoegd. Als de hypothese te algemeen blijkt wordt hij gespecificeerd.

Zowel Shapiro als Plotkin gebruiken een monotone logika als formalisme. Een monotone logika kent het volgende probleem: als een theorie gefalsificeerd wordt is de theorie waardeloos geworden. Er zijn verschillende mogelijkheden om dit probleem binnen de logika te omzeilen. Een mogelijkheid is gebruik te maken van een niet-monotone logika. Shapiro maakt gebruik van een andere mogelijkheid: een RMS die een inkonsistent systeem 'repareert'.

Hierbij kunnen twee vragen worden gesteld. De eerste is hoe een *conjecture* kan worden verzwakt als hij te sterk blijkt, als een onware uitspraak wordt geïmpliceerd. De tweede is hoe een *conjecture* kan worden versterkt als hij te zwak blijkt, als een feit niet kan worden afgeleid.

Als een *conjecture* te sterk is wordt er een hypothese uit verwijderd. Welke hypothese uit de *conjucture* moet worden geselekteerd is een probleem, ook in de wetenschap.

> ... it has to be admitted that we can often test only a large chunk of theoretical system, and sometimes perhaps only the whole system, and that, in these cases, it is sheer guesswork which of its ingredients should be held responsible for any falsification. – Popper [56, p. 14]

De hypothese die moet worden verwijderd wordt in het model bepaald door de *backtracing algorithm*. Door het uitvoeren van een eindige reeks vragen aan een orakel wordt er één hypothese uit de *conjecture* geselekteerd. De algoritme levert een tegenvoorbeeld van de geselekteerde hypothese, een uitspraak die logisch volgt uit de hypothese en eksperimenteel fout blijkt.

Als de hypothese is verwijderd kan de theorie te zwak zijn geworden. Er worden dan nieuwe hypothesen toegevoegd, gesuggereerd door verfijningsoperatoren (*refinement operators*), die gebruik maken van de verwijderde hypothesen. Er zijn verschillende operatoren, een algemene operator kan meer hypothesen afleiden maar is minder efficiënt dan een meer specifieke.

Shapiro heeft een lerend systeem geïmplementeerd, het `Model Inference System` (`MIS`). Het formalisme dat hij gebruikt is de Horn-clause logika, net als Muggleton en Buntine.

In het `MIS` is, net als in `CIGOL`, een besturingsmechanisme van buiten de logika toegevoegd, het orakel. Het verschil is dat in `MIS` alleen atomaire uitspraken en in `CIGOL` hele hypothesen worden getest.

Shapiro bewijst dat `MIS` in de limiet elk model kan identificeren. MIS is minder efficiënt dan `CIGOL`; er zijn veel meer voorbeelden nodig om een gezochte hypothese te vinden.

Een ander nadeel is dat het niet mogelijk is de taal uit te breiden, hetgeen bij Muggleton en Buntine wel mogelijk is:

> It should be noted that the assumption that the hypothesis language is fixed avoids one of the major problems in scientific discovery, which is the invention of a new conceptual scheme. In Kuhnian terminology, the model inference problem resembles more the puzzle-solving activity of "normal science" than the search for new conceptual schemes that is characterisic of "paradigm-shifts" periods. [56, p. 3]

Hoofdstuk 3

Logika

3.1 Inleiding

Logika is een deelgebied van de wiskunde en één van de oudste formalismen voor het representeren en manipuleren van informatie; de klassieke logika stamt uit de tijd van de oude Grieken. Aristoteles beschreef reeds een logika, gebaseerd op *syllogismen*, in zijn *prior analytics*. Maar het duurde tot de negentiende eeuw tot de logika werd geformuleerd als een strenge wiskundige wetenschap door theoretici als Boole en Frege.

Logika wordt ook wel gezien als de studie naar korrekte gevolgtrekkingen [50, p. 30]. Boole noemde zijn baanbrekende boek uit 1854 "*An investigation of the laws of thought*". Het weergeven van gevolgtrekkingen in een natuurlijke taal is problematisch door de ambigue struktuur ervan. Door gevolgtrekkingen uit de natuurlijke taal te formaliseren kan de korrektheid ervan worden bewezen. Dit is uiteraard alleen mogelijk als aan de kennisrepresentatie-hypothese is voldaan, dus als kennis tot atomen gereduceerd kan worden.

In de middeleeuwen probeerde men met behulp van logika te bewijzen dat er een God bestaat. Een voorbeeld hiervan zijn de ontologische Godsbewijzen waarbij men uitgaat van een bepaald begrip van God en dan tracht aan te tonen dat er in de werkelijkheid iets mee moet korresponderen. Zo schrijft Thomas van Aquino:

> Het tweede bewijs voor het bestaan van God berust op het begrip van de werkende oorzaak. Het is immers een feit dat er in de stoffelijke wereld een aaneenschakeling is van werkende oorzaken. Men vindt echter nergens een wezen – en het is ook onmogelijk er één te vinden – dat de werkende oorzaak van zichzelf zou zijn, *want* dan zou het eerder dan zichzelf moeten zijn en dit is uitgesloten. ... Wij moeten dus een eerste werkende oorzaak aannemen, die door allen God genoemd wordt. [21, p. 14]

Aan het begin van deze eeuw stond de logika in de belangstelling door het werk van Russell en Whitehead. Zij probeerden te bewijzen dat de getaltheorie uit de wiskunde af te leiden is van een eindig aantal axiomata en wel zonder tegenstrijdigheden. Peano had in de jaren '80 van de vorige eeuw een axiomatisatie van de getaltheorie in een eerst-orde logika beschreven. Russell en Whitehead beschreven in de *principia mathematica* de *typenleer*. Gödel bewees echter in 1931 met de onvolledigheidsstelling dat het onmogelijk is om de getaltheorie af te leiden uit een eindige axiomatisatie en dat de typenleer en de axiomatisatie van Peano dus niet volledig kunnen zijn [17, p. 24].

Na de Tweede Wereldoorlog werden vele nieuwe logika's ontwikkeld. Dit waren vooral niet-monotone logika's, die een grotere uitdrukkingskracht hebben dan de klassieke logika's.

Met de opkomst van de komputer beleefde de logika als kennisrepresentatieformalisme een bloei. Hierbij wordt gebruik gemaakt van minder krachtige, maar efficiëntere logika's.

Als kennisrepresentatie-formalisme wordt meestal de clause logika of de Hornclause logika gebruikt. De clause logika en de Horn-clause logika zijn deelverzamelingen van de klassieke eerste-orde predikatenlogika. In de volgende paragraaf worden deze logika's informeel geïntroduceerd, in de daarop volgende paragrafen worden zij formeel besproken. Deze formele beschrijving is noodzakelijk voor een analyse van de eigenschappen van logische generalisatie en induktief leren in de volgende hoofdstukken.

3.2 Logisch systeem

Binnen een logika worden uitspraken formules genoemd, waarmee wordt benadrukt dat een uitspraak onafhankelijk van zijn betekenis kan worden gemanipuleerd. Een logisch systeem bestaat uit twee delen, de syntaks met een deduktietheorie en de semantiek. Een deduktietheorie geeft aan of een uitspraak afleidbaar is, de semantiek geeft aan of een uitspraak waar is.

Binnen een logika zijn een aantal toegestane (welgevormde) formules gedefinieerd. Dit zijn betekenisloze symboolreeksen. De verzameling van alle symbolen en van alle welgevormde formules is de taal van de logika.

De taal bepaalt het uitdrukkingsvermogen van een logika. Als de taal van een logika wordt uitgebreid – als er meer formules worden toegestaan – dan is het mogelijk meer uit te drukken.

Bij een logika hoort een deduktietheorie, bestaande uit axiomata en afleidingsregels. De axiomata zijn een deelverzameling van de taal van een logika, een aantal welgevormde formules. Met behulp van de afleidingsregels kunnen logische formules op grond van hun vorm worden gemanipuleerd.

De axiomata dienen als basis voor de verzameling stellingen afleidbaar binnen de logika. Stellingen zijn welgevormde formules die door toepassing van de afleidingsregels uitgaande van de axiomata verkregen kunnen worden. De afleiding zelf – gebruikte afleidingsregels – geven de redenering weer die nodig was voor het afleiden van de stelling.

Axiomata en afleidingsregels bepalen de eigenschappen van een logika, bijvoorbeeld of een logika monotoon of niet-monotoon is.

De syntaks van een logika is een betekenisloos stelsel van welgevormde, afleidbare formules. Een semantiek geeft aan iedere welgevormde formule uit de taal van de logika een betekenis, onafhankelijk van het feit of de formule afleidbaar is.

Een betekenisfunktie geeft een welgevormde formule een betekeniswaarde (waarheidswaarde). Een logika met de waarden *waar* en *onwaar* heet tweewaardig. De propositielogika, de eerste-orde predikatenlogika, de clause logika en de Horn-clause logika zijn tweewaardige logika's. Naast de tweewaardige logika's zijn er ook logika's met een semantiek die meerwaardig is. Meest voorkomend is een driewaardige semantiek: een formule is *waar*, *onwaar* of *onbepaald*.

Afleidbaarheid (binnen de syntaks) en waarheid (binnen de semantiek) kunnen worden vergeleken. Als alle stellingen *waar* zijn is de deduktietheorie *gezond* voor die semantiek. Als alle formules die *waar* zijn ook kunnen worden afgeleid, dan is de deduktietheorie *kompleet*.

Een algoritme dat, gegeven een formule, in een eindige tijd kan bepalen of de formule waar is of niet heet een beslissingsprocedure. Voor een *gezonde* en *komplete* deduktietheorie is dit ekwivalent met het bepalen of een formule afleidbaar is of niet. Een logika waar een beslissingsprocedure voor bestaat heet beslisbaar, een logika waar geen beslissingsprocedure voor bestaat heet onbeslisbaar.

3.2.1 Propositielogika

In de taal van de propositielogika kunnen atomaire uitspraken worden weergegeven en kunnen met behulp van konnektieven nieuwe uitspraken worden gekonstrueerd uit een aantal atomaire of samengestelde uitspraken. De uitspraken of formules worden proposities genoemd. Voorbeelden van proposities zijn:

- *Vanavond ga ik naar de kroeg of ga ik vroeg naar bed.*
- *Als ik ziek ben, ga ik naar de dokter.*
- *Als deze skriptie af is, ga ik op vakantie of in dienst.*

Een atomaire uitspraak wordt met een kleine letter genoteerd. Er zijn vijf konnektieven, de negatie $\neg$, de konjunktie $\wedge$, de disjunktie $\vee$, de implikatie $\rightarrow$ en de ekwivalentie (bi-implikatie) $\leftrightarrow$. De negatie staat voor *niet*, de konjunktie voor *en*, de disjunktie voor *of*, de implikatie voor *als ... dan ...* en de ekwivalentie voor *dan en slechts dan als*. Bovenstaande voorbeelden worden in de propositielogika weergegeven als[1]:

[1]Deze formules zijn geen stellingen, zij zijn slechts waar voor de interpretatie die eronder staat. Stellingen zijn voor iedere interpretatie waar.

- $p \vee q$

  $p =$ Vanavond ga ik naar de kroeg.

  $q =$ Vanavond ga ik vroeg naar bed.

- $p \rightarrow q$

  $p =$ Ik ben ziek.

  $q =$ Ik ga naar de dokter.

- $p \rightarrow (q \vee r)$

  $p =$ Deze skriptie is af.

  $q =$ Ik ga op vakantie.

  $r =$ Ik ga in dienst.

In de semantiek kan iedere atomaire formule waar of onwaar zijn en kan de waarheidswaarde van een samengestelde formule worden afgeleid uit de proposities waaruit hij is opgebouwd. De waarheidswaarde van een formule is een funktie van de waarheidswaarden van de atomaire proposities die er in voorkomen. De waarheidswaarden van bovenstaande proposities zijn funkties van de waarheidswaarden van respektievelijk p en q, p en q en p, q en r.

Om de waarheidswaarden van een formule te kunnen bepalen, afhankelijk van de waarden van de atomaire proposities waaruit hij is opgebouwd, wordt er voor ieder konnektief een waarheidstabel opgesteld. De waarheidstabel voor bijvoorbeeld het konnektief $\vee$ (in een formule $p \vee q$) heeft de waarheidswaarde *waar* voor de kombinaties van p en q waarbij òf p òf q (of beide) de waarheidswaarde *waar* heeft. De waarheidstabel van het konnektief $\rightarrow$ (in een formule $p \rightarrow q$) heeft de waarheidswaarde *waar* voor kombinaties van p en q waarbij òf p de waarheidswaarde *onwaar* òf q de waarheidswaarde *waar* heeft.

Als twee formules dezelfde waarheidstabellen hebben, dan zijn zij ekwivalent. Zo is bijvoorbeeld $p \rightarrow (q \vee r)$ ekwivalent met $(\neg q \wedge \neg r) \rightarrow \neg p$, het laatste voorbeeld betekent hetzelfde als: *als ik niet in dienst ga en niet op vakantie ga, dan is deze skriptie nog niet af.*

De propositielogika is beslisbaar, want door bovenstaande werkwijze met waarheidstabellen toe te passen kan in eindige tijd worden bepaald of een formule de waarheidswaarde *waar* of *onwaar* heeft.

Er zijn twee afleidingsregels voor de propositielogika die (met een drietal axiomata) *gezond* en *kompleet* zijn, namelijk *uniforme substitutie* en *modus ponens*. *Uniforme substitutie* is het vervangen van alle atomaire proposities in een formule door willekeurige andere formules, waarbij dezelfde propositie door dezelfde formule vervangen moet worden. *Modus ponens* is het afleiden van q uit p en $p \rightarrow q$.

3.2.2 Eerste-orde predikatenlogika

De eerste-orde predikatenlogika is een uitbreiding op de propositielogika. De taal is uitgebreid met predikaten, variabelen, konstanten, funkties en kwantoren. De logika is gebaseerd op de verzamelingenleer uit de wiskunde, het is mogelijk te spreken over elementen van een verzameling.

De konstanten verwijzen naar een specifiek element, de variabelen verwijzen naar een willekeurig element en de funkties leggen een relatie tussen konstanten, variabelen of funkties. Zo kunnen oneindig komplekse relaties worden weergegeven.

De atomaire proposities uit de propositielogika zijn vervangen door atomen, een relatie tussen konstanten, variabelen en funkties. Atomen hebben net als proposities binnen de propositielogika de waarheidswaarde *waar* of *onwaar*.

De kwantor $\forall$ representeert de relatie *voor alle* ... *geldt dat* ... en de kwantor $\exists$ representeert *er is een* ... *zodanig dat* ....

Een voorbeeld is de verzameling van mensen met de funkties *vader van* en *moeder van*. Een konstante verwijst naar een specifiek persoon, bijvoorbeeld naar Matt Johnson, en een variabele verwijst naar een willekeurig persoon. Met behulp van funkties kunnen relaties worden vastgelegd, zo kan worden verwezen naar de vader van Matt Johnson, de moeder van zijn grootvader, naar de vader van een willekeurig persoon, etc. Met behulp van de universele kwantor kan nu de uitspraak *alle vaders zijn mannen* worden geformaliseerd en met behulp van de existentiele kwantor de uitspraak *Matt Johnson heeft een vader*.

Het volgende klassieke probleem uit de *prior analytics* van Aristoteles kan wel worden weergegeven in de eerste-orde predikatenlogika maar niet in de propositielogika.

- Aanname: Alle mensen zijn sterfelijk;
- Aanname: Socrates is een mens;
- Konklusie: Socrates is sterfelijk.

Dit is als een stelling te formaliseren binnen de eerste-orde predikatenlogika als één van de volgende twee formules:

- $(\forall x(P(x) \to Q(x))) \to (P(a) \to Q(a))$;
- $(\forall x(P(x) \to Q(x)) \land P(a)) \to Q(a)$;
- $P(x)$: x is een mens;
- $Q(x)$: x is sterfelijk;
- a: Socrates.

De eerste-orde predikatenlogika is niet beslisbaar: er bestaat geen algoritme die kan beslissen of een willekeurige formule *waar* of *onwaar* is. De methode van de propositielogika, het werken met waarheidstabellen, gaat niet meer op omdat de verzamelingen oneindig kunnen zijn. Zelfs met een eindige verzameling kunnen oneindig veel verschillende (geneste) funkties worden gekonstrueerd.

Dit staat bekend als de *onbeslisbaarheid van de eerste-orde predikatenlogika* (theorema van Church). Wel is er een algoritme die termineert als de formule inderdaad een logisch gevolg is, wat semi-beslisbaarheid wordt genoemd. Deze algoritme, de resolutie-algoritme van Robinson [52], maakt gebruik van clause logika.

3.2.3 Clause logika

Clause logika is een deelverzameling van de eerste-orde predikatenlogika, waarin geen kwantoren en alleen de konnektieven $\vee$, $\wedge$ en $\neg$ in het alfabet voorkomen.

Alle wff binnen de eerste-orde predikatenlogika kunnen worden herschreven naar wff binnen de clause logika zonder informatieverlies. Een wff uit de eerste-orde predikatenlogika is precies dan waar als de korresponderende wff uit de clause logika waar is. Het uitdrukkingsvermogen van de clause logika is gelijk aan het uitdrukkingsvermogen van de eerste-orde predikatenlogika.

Het voorbeeld in de vorige paragraaf kan vertaald worden naar een stelling binnen de clause logika, die duidelijk minder inzichtelijk is dan de stelling binnen de eerste-orde predikatenlogika:

- $(P(x) \vee \neg P(a) \vee Q(a)) \wedge (\neg Q(x) \vee \neg P(a) \vee Q(a))$

Een bewijs binnen de clause logika is niet het opstellen van een stelling, maar het afleiden van een konklusie uit een aantal aannamen (wat logisch gezien ekwivalent is). De afleidingsregel die hierbij gebruikt wordt is de resolutieregel. Het voorbeeld wordt dan:

- Aanname: $(\forall x)(P(x) \rightarrow Q(x))$;
- Aanname: $P(a)$;
- Konklusie: $Q(a)$.

3.2.4 Horn-clause logika

Horn-clause logika is een deelverzameling van de clause logika. Niet alle redeneringen uit de clause logika kunnen worden weergegeven.

De motivatie voor een eenvoudiger logika dan de clause logika (en de eerste-orde predikatenlogika) is dat de resolutiealgoritme voor de Horn-clause logika minder kompleks is. Zij is eenvoudiger te implementeren en is veel efficiënter. Een andere motivatie is dat de semantiek veel eenvoudiger is.

Horn-clause logika beperkt zich tot definiete informatie. Bij definiete informatie kan uit een verzameling premissen een ondubbelzinnige konklusie worden getrokken.

De uitspraak *als ik klaar ben met deze skriptie ga ik op vakantie of in dienst* kan niet worden geformaliseerd in de Horn-clause logika, want uit de premisse *ik ben klaar met mijn skriptie* kan geen ondubbelzinnige konklusie worden getrokken: het is mogelijk dat ik in dienst ga, maar het is ook mogelijk dat ik op vakantie ga.

3.3 Eerste-orde predikatenlogika

Eerst zullen de verschillende onderdelen van de eerste-orde predikatenlogika – de taal, de deduktietheorie en de semantiek – formeel worden besproken. Vervolgens wordt de relatie tussen de deduktietheorie en de semantiek gelegd en wordt de beslisbaarheid besproken.

3.3.1 Taal

Definitie 1 *Taal Een taal bestaat uit:*

- *Een alfabet van alle toegestane symbolen;*
- *Een verzameling van alle symboolreeksen gedefinieerd op dit alfabet.*

Definitie 2 *Alfabet Het alfabet van een logika is een verzameling van symbolen. Deze symbolen worden ook wel de primitieven van de taal genoemd.*

Definitie 3 *Alfabet van de eerste-orde predikatenlogika De taal van de eerste-orde predikatenlogika heeft als primitieven:*

- *Voor ieder natuurlijk getal* n (≥ 1) *een aftelbaar oneindige verzameling van* n*-plaatsige predikaatsymbolen, genoteerd met hoofdletters:* P, P_1, $\ldots$, Q, R, $\ldots$;
- *Voor ieder natuurlijk getal* n (≥ 1) *een aftelbaar oneindige verzameling van* n*-plaatsige funkties, genoteerd met letters in het midden van het alfabet:* f, f_1, $\ldots$, g, h, $\ldots$;
- *Een aftelbaar oneindige verzameling van variabelen, genoteerd met de laatste letters van het alfabet:* v, v_1, $\ldots$, w, x, $\ldots$;
- *Een aftelbaar oneindige verzameling van konstanten*[2]*, genoteerd met de eerste letters van het alfabet:* a, a_1, ..., b, c, ...;
- *De konnektieven:* $\neg$ *(negatie),* $\vee$ *(disjunktie),* $\wedge$ *(konjunktie),* $\rightarrow$ *(implikatie),* $\leftrightarrow$ *(ekwivalentie);*
- *De kwantoren:* $\forall$ *(voor alle),* $\exists$ *(er bestaat);*
- *De leestekens: (,) en ,.*

De taal bestaat uit een oneindige verzameling van konstanten, funkties en predikaten, waarmee oneindige verzamelingen en oneindig veel verschillende relaties kunnen worden gerepresenteerd. Bij toepassing van deduktie zijn meestal slechts eindige verzamelingen of een eindig aantal relaties interessant. Het alfabet van de taal kan dan worden beperkt tot een eindige verzameling van konstanten, funkties en predikaten (de niet-logische symbolen).

[2] Een konstante wordt ook genoteerd met een 0-plaatsige funktie.

Definitie 4 *Term Een term wordt rekursief gedefinieerd:*

- *Een konstante is een term;*
- *Een variabele is een term;*
- *Laat* t_i $(1 \leq i \leq n)$ *een aantal termen en* f *een* n*-plaatsig funktiesymbool zijn. Dan is* $f(t_1, \ldots, t_n)$ *ook een term.*

Definitie 5 *Atoom Laat* P *een* n*-plaatsig predikaatsymbool en* t_i $(1 \leq i \leq n)$ *een aantal termen zijn. Dan is* $P(t_1, \ldots, t_n)$ *een atoom.*

Definitie 6 *Wff Bepaalde reeksen van de symbolen uit het alfabet gelden als wel-gevormde formules (well-formed formulae, wff), genoteerd met kleine letters uit het Griekse alfabet:* α, α_1, $\ldots$, β, $\ldots$. *De wff worden, net als de termen, rekursief gedefinieerd:*

- *Een atoom is een wff;*
- *Als* α *een wff is, is* $\neg\alpha$ *dat ook;*
- *Als* α *en* β *wff's zijn, dan is* $(\alpha \vee \beta)$ *dat ook.*
- *Als* α *een wff is en* x *een variabele, dan is* $(\forall x)\alpha$ *dat ook.*

Voorbeelden van termen, atomen en wff zijn:

- Termen: x, a, $f(x, y)$, $f(x, g(y))$, $f(x, g(f(x, x)))$, $\ldots$
- Atomen: $P(x, y, y)$, $P(a, y, y)$, $P(f(x, y), y, y)$, $P(f(x, g(y)), u, a)$, $Q(f(a))$, $Q(f(x, g(f(x, x))), y, y)$, $\ldots$
- Wff: $(P(x) \vee Q(y))$, $(\forall x)P(x)$, $(\forall x)(\forall y)(P(y) \vee \neg Q(x))$, $\ldots$

De formules zijn vaak onoverzichtelijk, om de formules leesbaarder te maken worden er nog een aantal konnektieven en een kwantor gedefinieerd. Zij zouden ook direkt in de definitie van een wff kunnen worden opgenomen, zoals verschillende auteurs doen, maar om te benadrukken dat zij niets aan het uitdrukkingsvermogen toevoegen is dat hier niet gedaan.

Definitie 7 *Konnektieven en kwantor De konnektieven* $\wedge$, $\rightarrow$ *en* $\leftrightarrow$ *en de kwantor* $\exists$ *worden gedefinieerd aan de hand van* $\vee$, $\neg$ *en* $\forall$*:*

- $(\alpha \wedge \beta) \stackrel{def}{=} \neg(\neg\alpha \vee \neg\beta)$;
- $(\alpha \rightarrow \beta) \stackrel{def}{=} (\neg\alpha \vee \beta)$;
- $(\alpha \leftrightarrow \beta) \stackrel{def}{=} ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$;
- $(\exists x)\alpha \stackrel{def}{=} \neg(\forall x)\neg\alpha$.

Een tweede manier om de formules leesbaarder te maken is een aantal haakjes te verwijderen. Een formule kan veel haakjes bevatten. Door een hierarchische orde van de konnektieven en kwantoren aan te geven kan een ekwivalente verkorte schrijfwijze ontstaan. Een tweetal haakjes mag worden weggelaten als links van het symbool '(' en rechts van het symbool ')' geen konnektieven met hogere prioriteit staan.

Definitie 8 *Binding order De volgende rangorde is de binding order van de wff van de eerste-orde predikatenlogika:*

$\neg, \forall, \exists$

$\vee$

$\wedge$

$\rightarrow, \leftrightarrow$

De volgende twee formules zijn ekwivalent, waarbij de tweede is herschreven met behulp van de twee bovengenoemde methoden:

- $(\forall x)\neg(\forall y)\neg(\neg(P(x) \vee Q(x)) \vee Q(y))$
- $\forall x \exists y (P(x) \vee Q(x) \rightarrow Q(y))$

Definitie 9 *Bereik Laat $(\forall x)\alpha$ een onderdeel van een wff zijn. Dan is α het bereik van x.*

Definitie 10 *Gebonden en vrije variabelen Een voorkomen van een variabele is gebonden als hij direkt achter een kwantor komt of in het bereik van eenzelfde variabele. Een vrije variabele is een variabele die niet door een kwantor gebonden is.*

In de wff $(((\forall x)P(x,y)) \vee Q(x))$ zijn de eerste twee x gebonden en zijn y en de derde x vrije variabelen. Het bereik van x is de wff $P(x,y)$.

Definitie 11 *Zin Een zin (een gesloten formule) is een wff zonder vrije variabelen.*

Definitie 12 *Universele afsluiting Laat α een wff en x_i $(1 \leq i \leq n)$ de vrije variabelen in α zijn. De universele afsluiting van α is dan gelijk aan $(\forall x_1) \ldots (\forall x_n)\alpha$.*

Bij een deduktie bestaat er meestal geen interesse in vrije variabelen; er wordt dan verondersteld dat alle vrije variabelen universeel gekwantificeerd zijn.

3.3.2 Deduktietheorie

Definitie 13 *Deduktietheorie Een deduktietheorie bestaat uit axiomata en afleidingsregels.*

De axiomata en afleidingsregels van de eerste-orde predikatenlogika worden in de volgende subparagrafen besproken, waarna dedukties binnen deze logika worden gedefinieerd.

Axiomata

Onder deduktie wordt hier het afleiden van stellingen verstaan, het afleiden van logische waarheden. De axiomata zijn logische axiomata, uitspraken die waar zijn voor iedere mogelijke interpretatie van de symbolen.[3]

Definitie 14 *Axiomata De axiomata zijn een verzameling van wff die de gereduceerde kennis van een logika representeren.*

Voor de axiomata van de eerste-orde predikatenlogika kunnen worden gegeven, moet eerst het begrip substitutie worden gedefinieerd.

Definitie 15 *Substitutie Laat α een wff zijn met o.a. de vrije variabelen v_i $(1 \leq i \leq n)$ en laat t_i $(1 \leq i \leq n)$ een aantal termen zijn. Een substitutie $\theta = \{v_1/t_1, \ldots, v_n/t_n\}$[4] is:*

- *de vervanging van alle gebonden variabelen in α, die voorkomen in een t_i, door een nieuwe variabele;*
- *de gelijktijdige vervanging in α van de vrije variabelen v_i door de termen t_i.*

De eerste vervanging zorgt ervoor dat er in de nieuwe termen geen variabelen door kwantoren kunnen worden gebonden die reeds in α voorkomen.

Definitie 16 *Axiomata van de eerste-orde predikatenlogika Het volgende rijtje wff is een mogelijke axiomatisatie van de eerste-orde predikatenlogika*[5]*. α, β en γ zijn willekeurige wff, δ is een wff waar de variabele x niet in voorkomt, x een variabele en t een term.*

- $\alpha \rightarrow (\beta \rightarrow \alpha)$
- $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$
- $(\neg\alpha \rightarrow \neg\beta) \rightarrow (\beta \rightarrow \alpha)$
- $(\forall x)\alpha \rightarrow \alpha\{x/t\}$
- $(\delta \rightarrow \beta) \rightarrow (\delta \rightarrow \forall x\beta)$

De eerste drie axiomata garanderen dat alle stellingen binnen de propositielogika ook stellingen binnen de eerste-orde predikatenlogika zijn (waarbij de proposities door predikaten zijn vervangen). Het vierde en vijfde axioma leggen het gebruik en de introduktie van de kwantoren vast.

[3] Een logische theorie bestaat uit een eerste-orde taal en een verzameling formulae. Bij het toepassen van deduktie bepalen de formulae de eigenschappen van de niet-logische symbolen. Soms worden de formulae echter ook tot de axiomata gerekend, de axiomata zijn dan niet-logisch. In een natuurkundig systeem zijn natuurwetten voorbeelden van niet-logische axiomata.

[4] Meestal wordt deze vorm van substitutie met rechte haken genoteerd.

[5] Er zijn verschillende manieren om de eerste-orde predikatenlogika te axiomatiseren. Welke keuze er wordt gemaakt is niet van belang, als dezelfde formules maar afleidbaar zijn. Uiteraard is deze keuze wel afhankelijk van de afleidingsregels.

Afleidingsregels

Definitie 17 *Afleiding Een afleidingsregel is een relatie R over de verzameling van de wff van een logika. Laat $\alpha_1, \ldots, \alpha_n$ en β wff zijn. Dan kan β afgeleid worden uit $\alpha_1, \ldots, \alpha_n$ desda*[6] $\langle \alpha_1, \ldots, \alpha_n, \beta \rangle \epsilon R$. *Deze relatie wordt genoteerd met:* $\vdash \alpha_1, \ldots, \alpha_n \Rightarrow \vdash \beta$.

Met het systeem van de eerste-orde predikatenlogika zijn twee afleidingsregels verbonden:

- Generalisatie.
- Modus ponens.

Definitie 18 *Generalisatie Laat α een wff en x een variabele zijn. Dan is de generalisatie afleidingsregel:* $\vdash \alpha \Rightarrow \vdash (\forall x)\alpha$.

Definitie 19 *Modus ponens Laat α en β wff zijn. Dan is de modus ponens afleidingsregel:* $\vdash \alpha, \alpha \rightarrow \beta \Rightarrow \vdash \beta$.

Modus ponens zorgt ervoor dat de axiomata ook als afleidingsregels werken. Het eerste axioma geeft bijvoorbeeld de afleidingsregel $\vdash \alpha \Rightarrow \vdash \beta \rightarrow \alpha$.

Deduktie

De deduktietheorie bepaalt welke wff afleidbaar zijn van een verzameling van wff.

Definitie 20 *Afleidbaar Laat T een verzameling van wff en α een wff zijn. α is afleidbaar van T, $T \vdash \alpha$, desda:*

- *α een axioma of een element van T is;*
- *α van de axiomata en T kan worden afgeleid met behulp van de afleidingsregels.*

Definitie 21 *Stelling Laat T de lege verzameling van wff en α een wff zijn. α is een stelling, $\vdash \alpha$, desda $T \vdash \alpha$.*

Stelling 1 *Deduktiestellingen Laat T een verzameling van wff en α en β wff zijn. Er geldt:*

- $T \vdash (\alpha \rightarrow \beta) \Rightarrow T, \alpha \vdash \beta$
- $T, \alpha \vdash \beta \Rightarrow T \vdash (\alpha \rightarrow \beta)$

Het gevolg van de deduktiestellingen is dat bij het probleem van afleidbaarheid – is een wff afleidbaar van een verzameling van wff – kan worden beperkt tot de stellingen van de logika. α is afleidbaar van de verzameling $\{\beta_1, \ldots, \beta_n\}$ desda $(\beta_1 \wedge \ldots \wedge \beta_n) \rightarrow \alpha$ een stelling is.

[6] Dan en slechts dan als.

Definitie 22 *Konsistent Laat T een verzameling van wff zijn. T is konsistent desda er geen wff α is zodanig dat $T \vdash \alpha \wedge \neg\alpha$. Als er wel zo'n α is dan is T inkonsistent.*

De verzameling van alle wff kan zo worden onderverdeeld in drie groepen: de afleidbare wff, de konsistente maar niet afleidbare wff en de inkonsistente wff.

3.3.3 Semantiek

Formules hebben de waarheidswaarde *waar* of *onwaar*. Deze waarheidswaarde is altijd gebonden aan het geven van een betekenis aan de symbolen van de taal. De betekenis van de kwantoren en konnektieven ligt vast, maar de betekenis van konstanten, funktie- en predikaatsymbolen kan variëren. Een interpretatie geeft deze niet-logische symbolen een betekenis.

Een interpretatie bestaat uit een domein waarover de variabelen lopen, een toevoeging van een element van dit domein aan alle konstanten, een toevoeging van een funktie op dit domein aan ieder funktiesymbool en een toevoeging van een relatie op het domein aan ieder predikaatsymbool. In het bijzonder zijn modellen voor een formule van belang, interpretaties waarvoor een formule waar is.

Interpretatie

Definitie 23 *Interpretatie Laat L is een eerste-orde logika zijn. Een interpretatie $I = \langle D, || \rangle$ van L bestaat uit:*

- *Een domein, de verzameling D;*
- *Een toevoeging van een element van D aan iedere konstante a uit L, genoteerd met $\mid a \mid$;*
- *Een toevoeging van een funktie van D^n naar D aan ieder n-plaatsig funktiesymbool f, genoteerd met $\mid f \mid$;*
- *Een toevoeging van een afbeelding van D^n naar $\{waar, onwaar\}$ aan ieder n-plaatsig predikaatsymbool P uit L, genoteerd met $\mid P \mid$.*

Definitie 24 *Toekenning aan variabelen Laat $I = \langle D, || \rangle$ een interpretatie zijn voor een taal L. Een toekenning aan variabelen met betrekking tot I is een toekenning van een element uit D aan iedere variabele in L.*

Definitie 25 *Toekenning aan termen Laat $I = \langle D, || \rangle$ een interpretatie zijn voor een taal L met een toekenning aan variabelen T_v. Een toekenning aan termen met betrekking tot I wordt gegeven door:*

- *variabelen: $\mid x \mid = T_v(x)$ als aangegeven in de toekenning aan variabelen;*
- *konstanten: $\mid a \mid$ als aangegeven in de interpretatie;*

- *funkties:* $| f(t_1, \ldots, t_n) | = | f | (| t_1 |, \ldots, | t_n |)$ *als* f *een* n*-plaatsig funktiesymbool is.*

Met deze voorbereidingen kunnen vervolgens aan de wff waarden worden toegekend.

Definitie 26 *Toekenning aan wff Laat* $I = \langle D, || \rangle$ *een interpretatie zijn voor een taal* L *met een toekenning aan termen. Een toekenning aan wff met betrekking tot* I *wordt gegeven door:*

- *Atomen: Een atomaire formule* $P(t_1, \ldots, t_n)$ *krijgt de waarde* waar *als* $| P | (| t_1 |, \ldots, | t_n |)$ waar *is;*

- *Konnektieven: De waarheidswaarde van* $\neg\alpha$ *en* $\alpha \vee \beta$ *volgt uit de waarheidstabellen;* $\neg\alpha$ *is* waar *desda* α onwaar *is,* $\alpha \vee \beta$ *is* waar *desda* α *of* β waar *is (of beide* waar *zijn). De waarheidswaarden van de overige konnektieven volgen uit de definities.*

- *Kwantoren: De waarheidswaarde van* $(\forall x)\alpha$ *is* waar *desda voor ieder element* $d \epsilon D$ *geldt dat* α *waar is, als we dezelfde interpretatie en dezelfde toekenning van variabelen nemen, met de uitzondering dat* d *aan* x *wordt toegekend. De waarheidswaarde van* $\exists$ *volgt weer uit de definitie.*

Laat het domein de verzameling van alle mensen zijn. a – een konstante – is in de interpretatie dan een specifiek persoon, x – een variabele – een willekeurig persoon. Laat P een twee-plaatsig predikaat zijn met de afbeelding van twee mensen naar *waar* als de eerste de vader van de tweede is. De wff $\alpha = (\forall x, y)(P(x, y) \to \neg P(y, x))$ is *waar* in deze interpretatie, want de vader van een persoon kan niet zijn zoon zijn. Kiezen we nu echter een andere interpretatie, bijvoorbeeld als domein de natuurlijke getallen en voor $P(x, y)$ de afbeelding x *is groter of gelijk aan* y, dan is de waarheidswaarde van α *onwaar.*

De waarheidswaarde van een gesloten formule is niet afhankelijk van de toekenning aan variabelen. Dit is de reden dat we bij een deduktie slechts zijn geïnteresseerd in gesloten formules.

Modellen

Definitie 27 *Model Laat* I *een interpretatie en* α *een wff zijn.* I *is een model voor* α, $I \models \alpha$, *desda* α *de waarheidswaarde* waar *heeft in* I. *Een model voor een verzameling van gesloten wff is een interpretatie waarin iedere wff van die verzameling de waarheidswaarde* waar *heeft.*

Definitie 28 *Geldig Laat* α *een wff zijn.* α *is geldig,* $\models \alpha$, *desda* α *waar is in iedere interpretatie. Zo'n wff wordt ook wel een tautologie genoemd.*

Definitie 29 *Vervulbaar Laat* α *een wff zijn.* α *is vervulbaar desda* α *een model heeft, dus desda* $\neg\alpha$ *niet geldig is.*

In een interpretatie is iedere wff waar of onwaar en over alle interpretaties is een wff geldig, niet geldig maar wel vervulbaar of onvervulbaar. Net als bij afleidbaarheid worden alle wff dus in drie groepen verdeeld.

Definitie 30 *Logisch gevolg Laat S een verzameling van gesloten wff en α een wff zijn. α is een logisch gevolg van S, $S \models \alpha$, desda alle modellen van S een model van α zijn, ofwel als α waar is in alle modellen van S.*

Het probleem bij het bepalen of een wff geldig is, is dat er oneindig veel interpretaties en modellen zijn. Daarom wordt er gezocht naar een interpretatie die een model is voor een wff desda die wff geldig is.

Een voorbeeld van een interpretatie met enkele goede eigenschappen is door Herbrand in de jaren dertig opgesteld. In een Herbrand interpretatie wordt iedere wff 'door zichzelf' geïnterpreteerd, het domein bestaat uit termen.

Als een wff een model heeft, hoeft zij echter nog geen Herbrand model te hebben [8, p. 320]. Dit geldt wel voor de wff van de clause logika. In paragraaf 3.4 zullen de Herbrand modellen formeel worden gedefinieerd.

3.3.4 Relatie tussen syntaks en semantiek

Er zijn nu twee verschillende begrippen ingevoerd, afleidbaarheid en logische implikatie. De relatie tussen beide wordt door de volgende twee belangrijke theorema gelegd, waaronder de beroemde volledigheidsstelling van Gödel:

Stelling 2 *Volledigheidsstelling van Gödel Laat T een verzameling van wff en α een wff zijn.*

- *De afleidingsregels zijn gezond: $T \vdash \alpha \Rightarrow T \models \alpha$.*
- *De afleidingsregels zijn kompleet: $T \models \alpha \Rightarrow T \vdash \alpha$.*

Afleidbaarheid en logische implikatie zijn dus ekwivalent; alles wat kan worden afgeleid is geldig en alles wat geldig is kan worden afgeleid. Een gevolg is dat konsistent en vervulbaar ook ekwivalente begrippen zijn.

3.3.5 Beslisbaarheid

In plaats van uit te gaan van de axiomata en proberen te bewijzen dat een wff kan worden afgeleid, kan ook van de wff worden uitgegaan en worden getracht te bewijzen dat hij in het systeem past. De eerste methode van redeneren is een vorm van *forward reasoning* en de laatste van *backward reasoning* (zie paragraaf 2.3.3).

Definitie 31 *Beslisbaar Laat L een logika zijn. L is beslisbaar desda er een algoritme bestaat dat in een eindige tijd bepaalt of een willekeurige wff α geldig is (wat voor een komplete deduktietheorie ekwivalent is met het bepalen of α afleidbaar is).*

Definitie 32 *Semi-beslisbaar Laat L een logika zijn. L is semi-beslisbaar desda er een algoritme bestaat dat, als α* geldig *(of afleidbaar) is, dit in een eindige tijd bepaalt.*

De propositielogika is beslisbaar. Door waarheidstabellen op te stellen kan in een eindige tijd van iedere wff bepaald worden of hij waar is of niet.

Iedere eerste-orde logika is niet beslisbaar, maar wel semi-beslisbaar. Een voorbeeld van een algoritme dat probeert te bewijzen of een wff afleidbaar is van een verzameling van wff en termineert als dit zo is, is de resolutie-algoritme van Robinson [52]. Hierbij wordt gebruik gemaakt van de de clause logika. In de volgende paragraaf wordt de resolutie-algoritme besproken.

3.4 Clause logika

De clause logika is een deelverzameling van de eerste-orde predikatenlogika. De eigenschappen van wff binnen de clause logika (afleidbaarheid en waarheid) zijn ekwivalent met deze eigenschappen van de korresponderende wff binnen de eerste-orde predikatenlogika.

Alle gesloten wff binnen de eerste-orde predikatenlogika kunnen met behulp van een eindig algoritme worden herschreven naar wff binnen de clause logika (hun clausale vorm), waarbij beide wff konsistent of beide wff inkonsistent zijn; het uitdrukkingsvermogen van beide logika's is gelijk. De clause logika heeft een duidelijkere struktuur waardoor het eenvoudiger is om te beslissen of een wff konsistent is.

In de drie volgende paragrafen zullen de taal, de deduktietheorie en de semantiek van de clause logika worden besproken.

De deduktietheorie bestaat uit slechts één afleidingsregel, resolutie, en geen axiomata.

De interpretaties binnen de semantiek beperken zich tot Herbrand interpretaties. Iedere wff heeft een model desda hij een Herbrand model heeft.

3.4.1 Taal

Definitie 33 *Alfabet Het alfabet van de clause logika bestaat uit de symbolen $\{$ en $\}$ en het alfabet van de eerste-orde predikatenlogika, zonder de kwantoren en met alleen de* negatie, disjunktie *en* konjunktie *als konnektieven.*

Definitie 34 *Literal, clausal form en theorie De volgende wff worden gedefinieerd:*

- *Een literal is een atoom (positieve literal) of een negatie gevolgd door een atoom (negatieve literal). Literals worden genoteerd met L, L_1, L_2, ...*

- *Een clausal form*[7] *is een disjunktie van een eindig aantal literals. Clausal forms worden genoteerd met* C, C_1, C_2, ..., D, ... *Een clausal form ziet er als volgt uit:*

  (3.1) $L_1 \vee L_2 \vee \ldots \vee L_n$

- *Een clause is een eindige verzameling van literals:*

  (3.2) $\{L_1, L_2, \ldots, L_n\}$

- *Een (logische) theorie form is een konjunktie van een eindig aantal clausal forms. Theorie forms worden genoteerd met* T, T_1, ... *Een theorie form ziet er als volgt uit:*

  (3.3) $C_1 \wedge C_2 \wedge \ldots \wedge C_n$

- *Een (logische) theorie is een eindige verzameling van clausal forms of clauses:*

  (3.4) $\{C_1, C_2, \ldots, C_n\}$

Definitie 35 *Wff De wff van de clause logika zijn de theorie forms van de eerste-orde predikatenlogika.*

Literals en clausal forms zijn speciale vormen van theorie forms en behoren dus ook tot de wff. Een literal, clausal form of theorie form korrespondeert met zijn *universele afsluiting* binnen de eerste-orde predikatenlogika, hiermee ligt de semantiek van de clause logika vast.

Laat een clausal form uit n literals met m universeel gekwantificeerde variabelen x_i bestaan. De clausal form korrespondeert dan met de volgende wff binnen de eerste-orde predikatenlogika:

(3.5) $(\forall x_1, x_2, \ldots, x_m)(L_1 \vee L_2 \vee \ldots \vee L_n)$

Als er n_1 negatieve en n_2 positieve literals in de clausal form voorkomen dan kan hij als volgt worden herschreven in de eerste-orde predikatenlogika, waarbij alle L_i positieve literals zijn:

(3.6) $(\forall x_1, x_2, \ldots, x_m)((L_1 \wedge L_2 \wedge \ldots \wedge L_{n_1}) \rightarrow (L_{n_1+1} \vee L_{n_1+2} \vee \ldots \vee L_{n_1+n_2}))$

Een clause representeert een verzameling van clausal forms, namelijk de disjunkties die met de elementen van de clause kunnen worden gevormd. Hierbij

[7] In de literatuur is het gebruikelijk om wat hier clausal form wordt genoemd, ook clause te noemen. Het onderscheid tussen clausal form en clause is door Nienhuys-Cheng [38] gemaakt en zal in de volgende hoofdstukken van belang zijn.
Er wordt gewoonlijk gesproken over de clausale vorm van een wff. Hier wordt echter de wff zelf bedoeld, daarom is *form* niet vertaald.

moeten alle elementen van de verzameling minstens één keer voorkomen in deze clausal forms. Deze clausal forms worden de clausal forms van de clause genoemd. Iedere clausal form behoort tot precies één clause.

Alle clausal forms van een clause zijn ekwivalent aan elkaar. Dit blijkt uit de axiomata en afleidingsregels van de eerste-orde predikatenlogika.

De clause $\{P(x), Q(x,y)\}$ heeft o.a. als clausal forms:

- $P(x) \vee Q(x,y)$
- $Q(x,y) \vee P(x)$
- $P(x) \vee P(x) \vee Q(x,y)$

Op identieke wijze representeert een logische theorie een verzameling van theorie forms.

Definitie 36 *Grond Een grondformule is een formule zonder variabelen.*

Definitie 37 *Complementaire literals Laat L een positieve literal zijn. Het paar $\{L, \neg L\}$ is (in willekeurige volgorde) een complementair paar.*

Definitie 38 *Lege clause De lege clause, $\Box$, is de clause met nul literals.*

Definitie 39 *Lege theorie De lege theorie, $\blacksquare$, is de theorie met nul clauses.*

Een theorie die bestaat uit een lege clause is nooit waar, want een clause is waar als minstens één literal waar is en de clause die geen literals bevat is dus nooit waar.

De lege theorie is altijd waar, want een theorie is waar als alle clauses waar zijn en de theorie die uit nul clauses bestaat is dus altijd waar.

De symbolen $\Box$ en $\blacksquare$ representeren respektievelijk ook de konstanten *onwaar* en *waar*.

Relatie met eerste-orde predikatenlogika

Definitie 40 *Skolem normaalvorm Laat α een wff in de eerste-orde predikatenlogika zijn. α is in de Skolem normaalvorm als α de universele afsluiting van een logische theorie is.*

Stelling 3 *Relatie Laat α een gesloten wff van de eerste-orde predikatenlogika zijn. α kan tot een wff β in de Skolem normaalvorm worden getransformeerd die een model heeft desda α een model heeft.*

Bewijs 1 *Zie [8, p. 319].*

Deze stelling betekent dat als we willen weten of een (gesloten) wff in de eerste-orde predikatenlogika vervulbaar of geldig is, we ook kunnen kijken of een korresponderende wff in de Skolem normaalvorm vervulbaar of geldig is. Het gevolg is dat alle uitspraken die in de eerste-orde predikatenlogika worden gedaan, ook in de clause logika kunnen worden gedaan; het uitdrukkingsvermogen van beide is gelijk.

Een wff binnen de eerste-orde predikatenlogika kan niet altijd naar een ekwivalente wff in de Skolem normaalvorm worden herschreven omdat existentiële kwantoren niet kunnen worden verwijderd in een transformatieproces. De truuk die wordt toegepast staat bekend als skolemiseren en komt erop neer dat, in plaats van te beweren dat er een objekt bestaat met bepaalde eigenschappen, ook een naam kan worden gekozen voor zo'n objekt. Deze naam is afhankelijk van de universeel gekwantificeerde variabelen in wiens bereik hij voorkomt [8, p. 319].

De wff $(\forall x)(\exists y)P(y,x)$ kan worden herschreven naar $(\forall x)P(f_y(x),x)$, waarbij f_y de naam is die aan het objekt wordt gegeven. Dat beide wff niet ekwivalent zijn blijkt uit het feit dat er andere symbolen in voorkomen: in de modellen van de eerste komt f_y niet voor.

3.4.2 Deduktietheorie

In de clause logika wordt gebruik gemaakt van een vorm van modus ponens die resolutie wordt genoemd. Bezie de volgende vier afleidingen. De eerste twee zijn modus ponens, de derde kan hiervan worden afgeleid. De eerste drie zijn afleidingen binnen de eerste-orde predikatenlogika. De vierde is resolutie binnen de clause logika:

- $\vdash \alpha, \alpha \rightarrow \beta \Rightarrow \vdash \beta$
- $\vdash \alpha, \neg\alpha \vee \beta \Rightarrow \vdash \beta$
- $\vdash \alpha \vee \gamma, \neg\alpha \vee \beta \Rightarrow \vdash \gamma \vee \beta$
- $\vdash C_1, C_2 \Rightarrow \vdash C_3$

Als α, β en γ in Skolem normaalvorm staan geldt dit nog niet voor $\alpha \vee \gamma$, $\neg\alpha \vee \beta$ en $\gamma \vee \beta$. De formules moeten nog worden herschreven door de universele kwantoren naar links te halen.

Stel dat α een positieve literal is, β en γ clausal forms zijn en alle drie de formules geen variabelen bevatten. Als C_1 en C_2 twee complementaire literals bevatten en geen variabelen, dan kunnen de derde en vierde afleiding aan elkaar gelijk zijn.

Resolutie is dan de afleidingsregel die de disjunktie neemt van twee clausal forms en hier het complementaire paar uit verwijdert. C_1 en C_2 worden geresolveerd en C_3 heet de resolvente.

Als er wel variabelen in C_1 en C_2 voorkomen dan zullen moeten zij worden herschreven tot de vorm $\alpha \vee \gamma$ en $\neg\alpha \vee \beta$ en $\gamma \vee \beta$ moet weer naar een clause te transformeren zijn. De oplossing hiervoor is *unifikatie*, dit wordt in de volgende paragraaf besproken.

Stel dat $\alpha = (\forall x)P(x)$ en γ geen variabelen bevat. Dan zijn de volgende drie ekwivalent en de derde is in Skolem normaalvorm:

- $(\alpha \vee \gamma)$
- $((\forall x)P(x) \vee \gamma)$
- $(\forall x)(P(x) \vee \gamma)$

De bovenstaande simpele versie van resolutie werkt alleen als C_1 en C_2 een complementair paar bevatten. Met behulp van het substitutie-axioma $(\forall x)\alpha \rightarrow \alpha\{x/t\}$ kunnen voor variabelen termen worden gesubstitueerd zodat zij dezelfde literal bevatten. Ook hierin voorziet *unifikatie*.

Als clause C_1 bijvoorbeeld literal $P(x)$ bevat en clause C_2 literal $\neg P(y)$, dan kan y door x worden vervangen en bevatten C_1 en C_2 een complementair paar. Als C_1 literal $P(x, f(y))$ bevat en C_2 literal $\neg P(a, x)$, dan is hier het complementair paar $\{P(a, f(y)), \neg P(a, f(y))\}$ van te konstrueren.

Unifikatie

Unifikatie is een methode van *pattern matching* om van twee literals een complementair paar te maken.

Unifikatie maakt gebruik van substitutie. De substitutie die bij de eerste-orde predikatenlogika is gedefinieerd (het gebruik van de universele kwantor) werkt niet in de clause logika, omdat die substitutie alleen vrije variabelen vervangt en wff binnen de clause logika geen vrije variabelen bevatten. Daarom wordt de definitie van substitutie aangepast (en wordt een kombinatie van gebruik en introduktie van de universele kwantor).

Definitie 41 *Substitutie Laat v_i $(1 \leq i \leq n)$ een aantal verschillende variabelen en t_i $(1 \leq i \leq n)$ een aantal termen zijn. Een substitutie $\theta = \{v_1/t_1, \ldots, v_n/t_n\}$ is de gelijktijdige vervanging van de variabelen v_i door t_i in een formule of term.*

Een substitutie heet een grondsubstitutie desda alle t_i grondtermen zijn en een herbenoeming van variabelen desda alle t_i verschillende variabelen zijn.

Definitie 42 *Alfabetische variatie Laat α en β twee wff zijn. α is een alfabetische variatie van β desda er een herbenoeming van variabelen $\theta = \{v_1/t_1, \ldots, v_n/t_n\}$ is zdd[8] $\alpha\theta = \beta$ en de variabelen in $\{t_i\}$ die niet voorkomen in $\{v_i\}$ ook niet voorkomen in α.*

$P(x, y, f(x, y, z))$ en $P(y, z, f(y, z, v))$ zijn alfabetische variaties. Alfabetische variaties zijn ekwivalent aan elkaar.

Definitie 43 *Apart gestandaardiseerd Laat α en β twee wff zijn. α en β zijn apart gestandaardiseerd als er geen variabele zowel in α als in β voorkomt.*

[8] Zodanig dat.

Laat α en β twee wff zijn. α en β kunnen apart worden gestandaardiseerd door de variabelen van α te herbenoemen.

Definitie 44 *Unificeerbaar Laat C_1 en C_2 twee apart gestandaardiseerde clausal forms zijn. C_1 en C_2 zijn unificeerbaar als er een substitutie θ is zdd $C_1\theta = C_2\theta$. De substitutie θ wordt de unifier genoemd.*

Definitie 45 *Samenstelling Laat $\theta_1 = \{x_1/s_1, \ldots, x_n/s_n\}$ en $\theta_2 = \{y_1/t_1, \ldots, y_m/t_m\}$ twee substituties zijn. De samenstelling van θ_1 en θ_2 is gelijk aan de substitutie*

(3.7) $\theta_1\theta_2 = \{x_1/s_1\theta_2, \ldots, x_n/s_n\theta_2, y_1/t_1, \ldots, y_m/t_m\}$

waaruit alle elementen y_i zijn verwijderd die ook voorkomen onder de x_i.

De samenstelling van $\theta_1 = \{x/y, y/f(a)\}$ en $\theta_2 = \{y/g(a), v/g(a)\}$ is gelijk aan $\theta_1\theta_2 = \{x/g(a), y/f(a), v/g(a)\}$.

Definitie 46 *Algemener dan Laat θ_1 en θ_2 twee substituties zijn. θ_1 is algemener dan θ_2 desda er een substitutie θ bestaat zdd $\theta_2 = \theta_1\theta$.*

Definitie 47 *Mgu Laat θ een unifier zijn. θ is de meest algemene unifier (most general unifier, mgu) als hij algemener is dan iedere andere unifier.*

Er is een algoritme om de mgu van twee clausal forms te bepalen, het unifikatie-algoritme [8, p. 323].

Resolutie

Voor de resolutieregel gegeven kan worden moet eerst het begrip faktor worden behandeld.

Definitie 48 *Droppen literal Laat $C = L_1 \vee \ldots \vee L_n$ een clausal form en L een literal L_i van C zijn. C zonder L, $C\backslash L$, is dan gelijk aan de clausal form $L_1 \vee \ldots \vee L_{i-1} \vee L_{i+1} \vee \ldots L_n$.*

Definitie 49 *Faktor Laat C een clausal form zijn en $S = \{L_1, \ldots, L_n\}$ een verzameling van minimaal twee literals in C die hetzelfde teken en predikaatsymbool hebben. $(C\backslash L_2, \ldots, L_n)\theta$ is een faktor van C desda θ de mgu van S is.*

Definitie 50 *Resolutie Laat C_1 en C_2 twee clausal forms zijn en C_1' en C_2' twee apart gestandaardiseerde faktors van C_1 en C_2 met een literal L_1 in C_1' en een literal L_2 in C_2' zdd $L_1\theta = \neg L_2\theta$ voor een mgu θ. De resolutie afleidingsregel is gelijk aan $\vdash C_1, C_2 \Rightarrow \vdash ((C_1'\backslash L_1) \vee (C_2'\backslash L_2))\theta$.*

Alle resolventen van de clausal forms van een clause vormen wederom clauses. Laat C_1 en C_2 twee (faktors van) clauses zijn die resolveren op literals L_1 en L_2. De resolvente is (waarbij \ de gebruikelijke betekenis heeft):

(3.8) $((C_1\backslash L_1) \cup (C_2\backslash L_2))\theta$

Definitie 51 *R^n Laat T een logische theorie zijn. $R^n(T)$ wordt rekursief gedefinieerd:*

- $R^0 = T$;
- R^n = *verzameling van alle resolventen van paren clauses in* $R^{n-1}(T)$.

Robinson [52] heeft bewezen dat T inkonsistent is desda er een n is zdd $R^n(T)$ de lege clause $\square$ bevat.

Definitie 52 *Resolutie De deduktietheorie resolutie is als volgt gedefinieerd:*

- *Geen axiomata;*
- *Alleen de afleidingsregel resolutie.*

De deduktietheorie resolutie heeft de volgende twee eigenschappen:

- Resolutie is gezond: als een wff kan worden afgeleid, dan is hij ook geldig.
- Resolutie is *refutation* kompleet: als een verzameling van wff onvervulbaar is kan de lege clause worden afgeleid.

Als een lege clause wordt afgeleid, dan is bewezen dat de wff niet konsistent is. Als wordt beslist of een wff α een logisch gevolg van de verzameling T is, wordt de negatie van α aan T toegevoegd en wordt getracht de lege clause af te leiden uit $T \cup \{\neg\alpha\}$. Dit is een bewijs uit het ongerijmde (reductio ad absurdum) en er wordt gebruik gemaakt van *backward reasoning*.

De resolutie-algoritme termineert als de verzameling inkonsistent is, als de wff een logisch gevolg is. Zij termineert bijvoorbeeld niet als de verzameling bestaat uit de wff $\alpha \vee \neg\alpha$, als wordt beslist of $\alpha \wedge \neg\alpha$ een logisch gevolg is van een willekeurige verzameling.

Resolutie kan dus niet alles afleiden, maar dit wordt over het algemeen niet als nadeel maar juist als voordeel gezien. Want de afleidingen die niet kunnen worden gemaakt zijn (semantisch) ook niet interessant. Voorbeelden hiervan zijn het feit dat het regent of niet regent (een tautologie) en de afleiding van het feit dat het regent dat het regent of één gelijk is aan twee. In feite levert resolutie een heuristiek op alle afleidingen.

3.4.3 Semantiek

Definitie 53 *Herbrand universum Laat L een eerste-orde taal zijn. Het Herbrand universum van L is de verzameling van alle termen die te konstrueren zijn uit de konstanten en funktiesymbolen in L. Als er geen konstanten in L voorkomen dan wordt de konstante 'a' toegevoegd.*

Definitie 54 *Herbrand basis Laat L een eerste-orde taal zijn. De Herbrand basis voor L is de verzameling van alle atomaire wff die kunnen worden gevormd door een predikaatsymbool uit L waarin termen uit het Herbrand universum zijn ingevuld.*

Definitie 55 *Herbrand interpretatie Laat L een eerste-orde taal zijn. De Herbrand interpretatie $I = \langle D, ||\rangle$ van L is gedefinieerd als:*

- *Het domein, de verzameling D, is het Herbrand universum;*
- *De toevoeging van een element van D aan iedere konstante a uit L is gelijk aan a, dus $\mid a \mid = a$;*
- *De toevoeging aan ieder n-plaatsig funktiesymbool f de funktie die aan de grondtermen $t_1, \ldots, t_n$ toevoegt $f(t_1, \ldots, t_n)$;*
- *De toevoeging van een afbeelding van D^n naar* {waar,onwaar} *aan ieder n-plaatsig predikaatsymbool P uit L.*

Stelling 4 *Herbrand model Laat S een eindige verzameling clausal forms (of clauses) zijn. Dan heeft S een model desda S een Herbrand model heeft.*

Bewijs 2 *Zie [27].*

Bij het bepalen of een logische theorie (een verzameling clausal forms) vervulbaar is, kan het onderzoek worden beperkt tot de Herbrand modellen.

3.5 Horn-clause logika

De Horn-clause logika is een deelverzameling van de clause logika en is de basis van (pure) Prolog. In [27] worden de eigenschappen van de Horn-clause logika uitgebreid besproken, hier wordt volstaan met een korte bespreking.

De taal, de deduktietheorie en de semantiek worden behandeld. De deduktietheorie bestaat uit een speciale resolutieregel en de semantiek beperkt zich wederom tot de Herbrand interpretaties.

3.5.1 Taal

Definitie 56 *Alfabet Het alfabet van de Horn-clause logika is gelijk aan het alfabet van de clause logika.*

Definitie 57 *Horn-clause De volgende wff worden gedefinieerd:*

- *Een Horn-clausal form is een clausal form met maximaal één positieve literal en een Horn-clause is een clause met maximaal één positieve literal.*
- *Een programma clause is een clause met precies één positieve literal. Dit korrespondeert in de eerste-orde predikatenlogika met de volgende formule, waarin alle L_i positieve literals zijn en alle variabelen universeel gekwantificeerd:*

  (3.9) $(L_1 \wedge L_2 \wedge \ldots \wedge L_n \rightarrow L_{n+1})$

- *Een goal clause is een clause met nul positieve literals. Dit korrespondeert in de eerste-orde predikatenlogika met de volgende formule, waarin alle L_i positieve literals zijn en alle variabelen universeel gekwantificeerd:*

  (3.10) $\neg(L_1 \wedge L_2 \wedge \ldots \wedge L_n)$

- *Een unit clause (feit) is een positieve literal.*

- *Een (logisch) programma form is een konjunktie van een aantal Horn-clausal forms en een (logisch) programma is een verzameling van Horn-clausal forms of Horn-clauses.*

Definitie 58 *Wff De wff van de Horn-clause logika zijn de logische programma forms.*

Een programma clause representeert definiete informatie. Uit de aanname van een aantal negatieve literals wordt de ondubbelzinnige konklusie van de positieve literal getrokken.

Definitie 59 *Rekursie Laat C een Horn-clausal form zijn. C is een rekursieve Horn-clausal form als C uit een positieve en negatieve literal bestaat die met hetzelfde predikaatsymbool beginnen en een partieel-rekursieve Horn-clausal form als C een een positieve en negatieve literal bevat die met hetzelfde predikaatsymbool beginnen.*

We kunnen niet meer alles uitdrukken in de Horn-clause logika, maar nog wel alle (partieel-)rekursieve funkties (met behulp van (partieel-)rekursieve clausal forms), wat voor algoritmische doeleinden voldoende is [8, p. 319].

3.5.2 Deduktietheorie

Definitie 60 *Lineaire resolutie Lineaire resolutie is een opvolging van resolutiestappen waarbij elke stap de resolvente wordt geresolveeerd met een clause.*

Definitie 61 *Selektieregel Een selektieregel is een funktie van goals naar atomen, zdd het atoom toegevoegd aan een goal altijd één van de premissen is van die goal. [8, p. 323]*

Een voorbeeld van een selektieregel is altijd de meest linkse van de premissen te kiezen, dit is de Prolog selektieregel.

SLD-resolutie is lineaire resolutie met behulp van definiete (Horn-)clauses en een selektieregel.

Definitie 62 *Sld-afleiding Laat P een logisch programma zijn en N een goal. Een SLD-afleiding is de reeks (N_i, C_i, θ_i) zdd:*

- $N_0 = N$

- N_{i+1} = *de resolvente van N_i en C_i waarbij:*

1. *C_i een alfabetische variatie van een clause in P is;*
2. *θ_i de mgu van C_i en de geselekteerde atoom van N_i is;*
3. *C_i en N_i geen overeenkomstige variabelen hebben.*

Een SLD-ontkenning is een SLD-afleiding waarbij één van de N_i de lege clause is. De mgu's leveren een tegenvoorbeeld van N.

CWA

SLD-resolutie kan alleen beslissen op of een aantal positieve literals waar (afleidbaar) zijn. De vraag is hoe negatieve informatie uit een programma gededuceerd kan worden. Een mogelijke oplossing is de deduktietheorie van de logika uit te breiden met de CWA afleidingsregel.

Definitie 63 *CWA De closed world assumption is de volgende afleidingsregel: als α geen logische gevolg is van een programma, dan is $\neg\alpha$ dat wel.*

Dit wordt ook wel 'negation as failure' genoemd. Deze afleidingsregel is niet-monotoon [27, p. 62]. In de rest van deze skriptie zal hier geen gebruik van worden gemaakt.

3.5.3 Semantiek

Stelling 5 *Doorsnede Herbrand modellen Laat P een logisch programma zijn. Dan is de doorsnede van alle Herbrand modellen voor P weer een Herbrand model voor P.*

Voor clausal forms geldt dit niet [8, p. 320]. Dit wordt veroorzaakt door de ondubbelzinnigheid van Horn-clausal forms (en niet van clausal forms).

Definitie 64 *Kleinste Herbrand model Laat P een logisch programma zijn. Het kleinste Herbrand model voor P, M_p, is de doorsnede van alle Herbrand modellen voor P.*

Stelling 6 *Kleinste Herbrand model Laat P een logisch programma zijn. Het kleinste Herbrand model voor P is gelijk aan de verzameling grondatomen die een logisch gevolg van P zijn.*

Bewijs 3 *Zie [8, p. 320].*

Definitie 65 *T Laat I een deelverzameling van B_p (de Herbrand basis) zijn. $T_p(I) = \{A \epsilon B_p \mid$ er is een* grondsubstitutie *van een programmaclause uit P van de vorm $A_1 \wedge \ldots \wedge A_n \rightarrow A$ en $A_i \epsilon I$ $\}$.*

Definitie 66 *$T\uparrow$ Laat P een logisch programma zijn. Dan is $T\uparrow$ gelijk aan:*

1. $T_p \uparrow 0 := \emptyset$
2. $T_p \uparrow (n+1) := T_p(T_p \uparrow n)$

3. $T_p \uparrow \infty := \cup\{T_p \uparrow n \mid n = 0, 1, \ldots\}$

Stelling 7 *Ekwivalentie semantiek en syntaks Laat P een logisch programma zijn. Dan geldt:*

(3.11) $M_p = T_p \uparrow \infty$

Dit betekent dat bij Horn-clauses het onderzoek zich kan beperken tot de kleinste Herbrand modellen, gerepresenteerd met een verzameling van grondliterals zonder een complementair paar. Laat M zo'n model zijn, L een literal, C een clause en P een logisch programma. Dan geldt:

- M is een model voor L desda alle grondliterals van L, verkregen door de variabelen in L te vervangen door een grondterm, een element van M zijn.
- M is een model voor C desda alle grondclauses van C, verkregen door de variabelen te vervangen door grondtermen, minstens één literal bevatten die in M zit.
- M is een model voor P desda M een model is voor alle clauses in P.

Procedurele semantiek

De stelling uit de vorige paragraaf levert een *forward reasoning* methode om te bewijzen of een (grond) feit afleidbaar is uit een logisch programma, namelijk het iteratief berekenen van $T_p \uparrow n$.

Efficiënter is de *backward reasoning* methode, het toevoegen van een goal aan het logische programma en met behulp van resolutie de lege formule $\Box$ proberen af te leiden (net als binnen de clause logika).

Stelling 8 *Gezond Laat P een programma en N een goal zijn. Als er een sld-ontkenning bestaat met antwoord-substitutie θ dan geldt $P \models \neg N\theta$.*

Stelling 9 *Kompleet Laat P een programma en N een goal zijn. Als $P \cup N$ onvervulbaar is, is er een sld-ontkenning van $P \cup N$.*

De definitie van T_p geeft aan wat de procedurele interpretatie van de Horn-clause logika is. De negatieve literals zijn aanroepen (premissen), de positieve is het resultaat (de konklusie). Zie [27, p. 2] en [8, p. 321].

Om dit te benadrukken worden de wff ook wel andersom geschreven, een programma clause wordt genoteerd met:

(3.12) $L_{n+1} \leftarrow L_n, \ldots, L_1$

Hoofdstuk 4

Generalisatie

4.1 Inleiding

Generalisatie is een belangrijke operatie bij induktief leren. Bij induktief leren wordt een aantal voorbeelden verklaard door een generalisatie van deze voorbeelden.

Er kan onderscheid worden gemaakt tussen *bottom up* generalisatie als in het model van Plotkin en *top down* generalisatie als in `MIS` van Shapiro. In het model van Plotkin wordt een hypothese steeds verder gegeneraliseerd, in het model van Shapiro wordt een zo algemeen mogelijke hypothese toegevoegd en wordt deze vervolgens gespecificeerd.

Het begrip generalisatie wordt ook in de wetenschap gebruikt om aan te geven dat een theorie meer uitspraken bevat. Deze definitie kan eenvoudig worden vertaald naar de logika.

De definitie van generalisatie is een bijzonder geval van het wiskundige begrip *ordening*. De wiskundige terminologie kan gebruikt worden bij analyse van (verschillende vormen van) generalisatie.

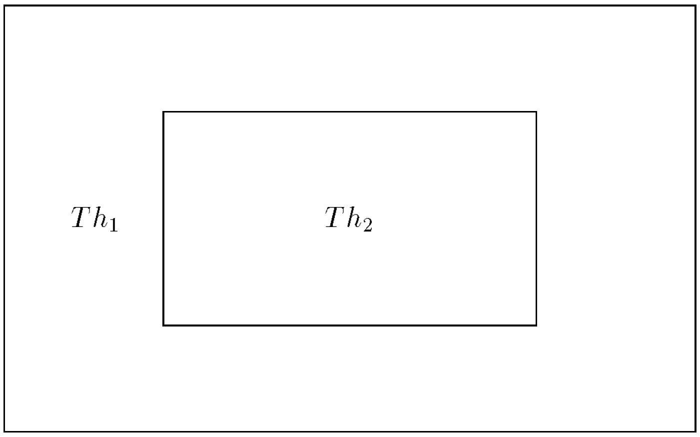
In dit hoofdstuk worden algemene definities van generalisatie behandeld. In het volgende hoofdstuk komen een aantal generalisatiemodellen van clausal forms zonder achtergrondtheorie aan de orde en vervolgens worden generalisaties met achtergrondtheorie bekeken. In hoofdstuk 7 wordt ingegaan op de toepassing van generalisatie bij induktief leren.

4.2 Algemener dan

4.2.1 Wetenschap

Een theorie in de wetenschap is een verzameling van uitspraken, dit kunnen feiten (het logische positivisme) of hypothesen (Popper) zijn (zie paragraaf 2.2.3). Theorieën kunnen met elkaar worden vergeleken, zie figuur 4.1 [35].

Definitie 67 *Algemener dan Laat Th_1 en Th_2 twee (wetenschappelijke) theorieën zijn. Th_1 is algemener dan Th_2 als alle uitspraken die in Th_1 zitten ook* *in Th_2 zitten, ofwel als Th_2 een deelverzameling van Th_1 is.*



Figuur 4.1: Th_1 is algemener dan Th_2

De wetenschappelijke theorieën Th_1 en Th_2 zijn ekwivalent als Th_1 algemener is dan Th_2 en Th_2 algemener is dan Th_1, ofwel als zij precies dezelfde uitspraken bevatten. Th_1 is strikt algemener dan Th_2 als Th_1 algemener is dan Th_2 en zij niet ekwivalent zijn.

In de terminologie van Popper (zie paragraaf 2.2.3) is een theorie strikt algemener desda zij meer uitsluit maar wel dezelfde uitspraken bevat. Alle uitspraken in Th_1 die niet in Th_2 zitten geven nu falsifikatoren voor de theorie, namelijk de ontkenningen van deze uitspraken. Een theorie is algemener als zij een hogere *verisimilitude* heeft.

Een theorie die de uitspraak "*Alle mensen zijn sterfelijk*" bevat is algemener dan de theorie die bestaat uit de uitspraak "*Socrates is sterfelijk*", want de eerste theorie bevat ook de uitspraak "*Socrates is sterfelijk*".

4.2.2 Logika

Als dit begrip van *algemener dan* in logische termen wordt vertaald, ligt het voor de hand een (wetenschappelijke) theorie te formaliseren als een model, de uitspraken van de theorie zijn ware formules in het model en alle andere uitspraken zijn onware formules.

Het probleem hierbij is dat iedere onware formule in het model een ware tegenhanger heeft, namelijk zijn negatie. Als een model dus een ware formule bevat die onwaar is in een ander model, dan is de negatie onwaar in het model maar waar in het andere model.

Definitie 68 *Algemener dan Laat M_1 en M_2 twee modellen zijn. M_1 is algemener dan M_2, $M_1 \geq_a M_2$, desda voor alle "positieve" α waarvoor geldt $M_2 \models \alpha$ ook geldt $M_1 \models \alpha$.*

In een logika is een theorie geaxiomatiseerd, een logische theorie is een verzameling van uitspraken (formules) die geldig zijn en waar alle geldige uitspraken van kunnen worden afgeleid. Formules zijn geldig, vervulbaar of onvervulbaar, zodat het bovenstaande probleem niet meer opgaat. De definitie van *algemener dan* kan eenvoudig worden vertaald naar logische theorieën.

Definitie 69 *Algemener dan Laat T_1 en T_2 twee logische theorieën zijn. T_1 is algemener dan T_2, $T_1 \geq_a T_2$, desda voor alle wff α waarvoor geldt $T_2 \models \alpha$ ook geldt $T_1 \models \alpha$.*

Uit de definities in hoofdstuk 3 volgen de volgende ekwivalente definities van *algemener dan* voor de logische theorieën T_1 en T_2:

- $T_1 \geq_a T_2$ desda voor alle wff α waarvoor geldt $T_2 \vdash \alpha$ ook geldt $T_1 \vdash \alpha$.
- $T_1 \geq_a T_2$ desda alle modellen van T_1 ook een model van T_2 zijn.
- $T_1 \geq_a T_2$ desda alle wff α, die waar zijn in alle modellen van T_2, waar zijn in alle modellen van T_1.

Deze definitie stelt ons in staat het algemene generalisatieprobleem te beschrijven voor een logische theorie [12].

Definitie 70 *Model-afleidingsprobleem Laat T een logische theorie zijn. Het model-afleidingsprobleem is het vinden van een onbekend model M (of een verzameling van modellen) zdd T waar is in het model.*

Dit onbekende model M kan worden gerepresenteerd met een logische theorie T'. Het aantal keuzen voor T' is (vrijwel) onbeperkt. In de praktijk zal daarom meer informatie beschikbaar zijn over welke generalisaties moeten worden gekozen. In het hoofdstuk over induktief leren wordt hierop nader ingegaan.

4.2.3 Monotone logika

In een monotone logika geldt dat een theorie T_1 algemener is dan een theorie T_2 desda alle wff in T_2 waar zijn in alle modellen van T_1.

Stelling 10 *Algemener dan Laat T_1 en T_2 twee logische theorieën zijn in een monotone logika. T_1 is algemener dan T_2 desda voor alle wff α in T_2 geldt dat $T_1 \models \alpha$ (of $T_1 \vdash \alpha$).*

Bewijs 4 *Als alle wff die afleidbaar zijn van de wff van T_2 ook afleidbaar zijn van T_1, dan zijn de wff van T_2 dit ook (triviaal).*

Als alle wff van T_2 afleidbaar zijn van T_1, dan zijn ook alle wff die van T_2 afleidbaar zijn, afleidbaar van T_1. Dit volgt uit de monotoon-eis.

Merk op dat dit voor een niet-monotone logika niet geldt. Stel dat T_2 een deelverzameling is van T_1. Alle wff van T_2 zijn dan triviaal afleidbaar van T_1, maar het is mogelijk dat van T_2 een wff kan worden afgeleid die van T_1 niet kan worden afgeleid.

Stelling 11 *Deelverzameling Laat T_1 en T_2 twee logische theorieën zijn in een monotone logika. Als T_2 een deelverzameling is van T_1 dan is T_1 algemener dan T_2.*

Bewijs 5 *Als T_2 een deelverzameling is van T_1 dan volgt daar triviaal uit dat alle wff in T_2 afleidbaar zijn van T_1.*

Merk op dat dit een voldoende, maar geen noodzakelijke voorwaarde is: als T_2 geen deelverzameling is van T_1 dan kan T_1 nog wel algemener zijn dan T_2.

Het Socrates voorbeeld kan als volgt worden geformaliseerd in de eerste-orde predikatenlogika:

- $\{(\forall x)P(x)\} \geq_a \{P(a)\}$, want $\vdash (\forall x)P(x) \Rightarrow \vdash P(a)$ (substitutieaxioma).
- $P(x)$: x is sterfelijk;
- a: Socrates.

Het resultaat van het model-afleidingsprobleem wordt in een monotone logika gewoonlijk als uitgebreide logische theorie $T \cup H$ gerepresenteerd. H kan dan bijvoorbeeld een verzameling hypothesen zijn die een bepaald verschijnsel verklaart.

4.2.4 Axiomatisatie-probleem

Een axiomatisatie wordt vaak verondersteld eindig te zijn want in een implementatie is het werken met oneindige verzamelingen moeilijk. Hierbij treedt echter het probleem op dat een axiomatisatie neveneffekten kan hebben, namelijk dat een algemene stelling ook beperkingen oplegt aan konsistente formules. In logika-termen betekent dit dat axiomatisaties niet logisch ekwivalent zijn. Dit axiomatisatie-probleem wordt geïllustreerd aan de hand van een voorbeeld.

Laat M het model van alle niet-priemgetallen zijn, M_1 het model van alle even getallen groter dan twee en M_2 het model van negen en alle even getallen groter dan twee. M is algemener dan M_1 en M_2, want negen en alle even getallen groter dan twee zijn geen priemgetallen.

M is het model dat via het model-afleidingsprobleem uit een axiomatisatie van M_1 gevonden moet worden. Eerst wordt M_1 tot de logische theorie T geaxiomatiseerd, T wordt vervolgens gegeneraliseerd tot een uitgebreide theorie $T \cup H$ van M_2 en uiteindelijk hopen we een axiomatisatie voor M te bereiken.

In M_1 geldt dat als een getal de waarheidswaarde *waar* heeft, als het dus een even getal groter dan vier is, dat het getal plus twee ook de waarheidswaarde *waar* heeft. Dit geldt echter niet in M en M_2.

Het voorbeeld wordt formeel uitgewerkt. Laat s de opvolger-funktie zijn. Getallen worden dan gerepresenteerd als opvolgers van nul, afgekort m.b.v. $s^n(0)$, de n-de opvolger van nul. Laat $P(x)$ het atoom zijn met interpretatie *x is geen priemgetal*. De wff $\alpha = (\forall x)(P(x) \rightarrow P(s^2(x)))$ is konsistent in een axiomatisatie van M_1, echter niet in een axiomatisatie van M_2.

De logische theorieën T_1 en T_2 zijn twee axiomatisaties van M_1; T_1 een oneindige en T_2 een eindige axiomatisatie. Beide zijn niet logisch ekwivalent (zie [28]) maar M_1 is het kleinste Herbrand model van beide logische theorieën.

- $T_1 = \{P(s^4(0)), P(s^6(0)), P(s^8(0)), \ldots\}$
- $T_2 = \{P(s^4(0)), (\forall x)(P(x) \rightarrow P(s^2(x)))\}$
- $P(x)$: x is een niet-priemgetal

M_2 is wel een model van T_1, maar niet van T_2. Als T_2 wordt uitgebreid met het feit $P(s^9(0))$, dan worden ook alle oneven getallen groter dan negen afleidbaar.

Dit probleem kan in T_2 verholpen worden door een nieuw predikaat te introduceren of de logische theorie te wijzigen (*debuggen*). Dit lapwerk is echter een besturingsmechanisme van buiten de logika en moet indien mogelijk vermeden worden.

Het punt waar het om draait is dat het model-afleidingsprobleem iets anders is dan het vinden van een algemener model dan het (kleinste) Herbrand model van de logische theorie. Een axiomatisatie legt restrikties op aan de mogelijke generalisaties.

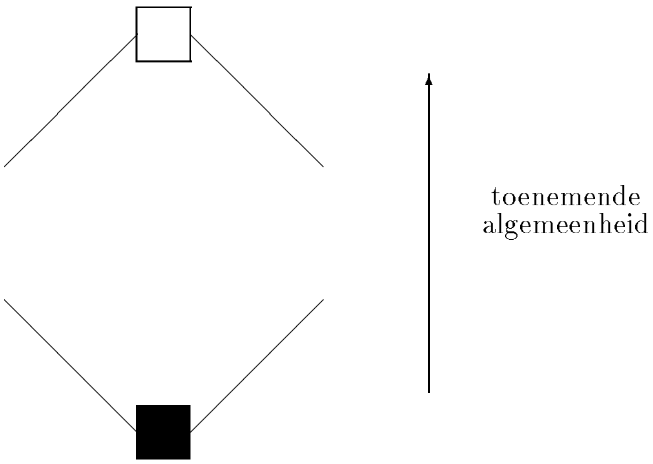
4.2.5 Formule

Een theorie is een verzameling van wff. Bij generalisatie wordt meestal niet de gehele theorie gegeneraliseerd, slechts één (of een aantal) wff uit de theorie. De theorie zonder deze wff wordt de achtergrondtheorie genoemd.

Vergelijk bijvoorbeeld de uitspraken $U_1 =$"*Het heeft geregend*" en $U_2 =$"*De straten zijn nat*". We zeggen dat U_1 strikt algemener is dan U_2, want als het heeft geregend dan zijn de straten nat maar andersom hoeft dit niet te gelden. Deze kennis bevindt zich in de achtergrondtheorie.

In de propositielogika kan dit voorbeeld worden geformaliseerd als:

- $\{p\} \geq_a \{q\}(p \rightarrow q)$;
- p: Het heeft geregend;
- q: De straten zijn nat.



Figuur 4.2: De generalisatietralie

In de clause logika is een logische theorie een verzameling van clausal forms, twee clausal forms kunnen dan vergeleken worden ten opzichte van een achtergrondtheorie:

Definitie 71 *Relatief algemener dan Laat* T *een theorie en* C_1 *en* C_2 *twee clausal forms zijn.* C_1 *is algemener dan* C_2 *relatief aan* T, $C_1 \geq_a C_2(T)$, *desda* $C_1 \cup T$ *algemener is dan* $C_2 \cup T$ *(wat ekwivalent is met algemener dan* C_2, *want* T *is immers triviaal afleidbaar).*

Twee clauses zijn ekwivalent, relatief aan een achtergrondtheorie, als zij allebei *waar* of *onwaar* zijn in alle modellen van de achtergrondtheorie.

4.3 Posets en tralies

Algemener dan is een ordening op de verzameling van ekwivalentieklassen van formules. In een ekwivalentieklasse zitten logische ekwivalente formules, ofwel de formules die waar zijn in dezelfde modellen.

De verzameling van ekwivalentieklassen met de *algemener dan* ordening vormt de tralie van figuur 4.2 [32], waarbij $\square$ de lege clause en $\blacksquare$ de lege theorie is. Het supremum van een aantal wff is de konjunktie van die wff en het infimum is de disjunktie ervan.

In deze paragraaf worden enige algemene definities bij een verzameling met een ordening gegeven.

4.3.1 Ordening

Definitie 72 *Ordening Laat* P *een verzameling met een betrekking* $\sqsubseteq$ *zijn.* $\sqsubseteq$ *is een ordening desda voor alle* x, y *en* z *in* P *geldt:*

1. $x \sqsubseteq x$ *(de betrekking is reflexief);*

2. *Als* $x \sqsubseteq y$ *en* $y \sqsubseteq x$ *dan volgt* $x = y$ *(de betrekking is antisymmetrisch);*

3. *Als* $x \sqsubseteq y$ *en* $y \sqsubseteq z$ *dan volgt* $x \sqsubseteq z$ *(de betrekking is transitief);*

P met $\sqsubseteq$ *wordt een partieel geordende verzameling (partially-ordered set, poset) genoemd.*

Definitie 73 *Vergelijkbaar Laat* P *met* $\sqsubseteq$ *een poset zijn en* x *en* y *twee elementen in* P. x *en* y *zijn vergelijkbaar desda* $x \sqsubseteq y$ *of* $y \sqsubseteq x$.

Definitie 74 *Totale ordening Laat* P *met* $\sqsubseteq$ *een poset zijn.* $\sqsubseteq$ *is een totale (een lineaire) ordening desda alle* x *en* y *in* P *vergelijkbaar zijn.*

Een voorbeeld van een totale ordening is de *groter dan of gelijk aan* ordening op natuurlijke getallen; van elk tweetal (verschillende) getallen kan worden gezegd welke het grootste is.

Definitie 75 *Maximum en minimum, maximaal en minimaal Laat* P *met* $\sqsubseteq$ *een poset zijn en* x *een element in* P. x *is het maximum (minimum) van* P *desda voor alle* y *in* P *geldt* $x \sqsubseteq y$ $(y \sqsubseteq x)$. x *is maximaal (minimaal) als voor alle* y *in* P *geldt: uit* $y \sqsubseteq x$ $(x \sqsubseteq y)$ *volgt* $x = y$.

Iedere eindige verzameling heeft maximale en minimale elementen. Voor oneindige verzamelingen geldt dit niet; bijvoorbeeld de relatie *groter dan* op een open interval van de reële getallen.

Definitie 76 *Bovengrens en ondergrens Laat* P *met* $\sqsubseteq$ *een poset zijn en* x *en* y *twee elementen in* P. x *is een bovengrens (ondergrens) van* y *desda* $x \sqsubseteq y$ $(y \sqsubseteq x)$. x *is een strikte bovengrens (ondergrens) desda* x *een bovengrens (ondergrens) van* y *is maar* y *geen bovengrens (ondergrens) van* x.

Definitie 77 *Supremum en infimum Laat* P *met* $\sqsubseteq$ *een poset zijn,* S *een deelverzameling van* P *en* x *een element van* P. x *is het supremum (infimum) van* S, $\sqcup S$ $(\sqcap S)$, *desda* x *het minimum (maximum) van de bovengrenzen (ondergrenzen) van* S *is. Het supremum (infimum) van twee elementen* x_1 *en* x_2 *wordt genoteerd met* $x_1 \sqcup x_2$ $(x_1 \sqcap x_2)$.

Het supremum (infimum) van het open interval $\langle 0, 1\rangle$ op de verzameling reële getallen met de ordening *groter dan of gelijk aan* is gelijk aan één (nul). Merk op dat nul en één geen elementen van het open interval zijn.

Definitie 78 *Oneindig stijgend en dalend Laat* P *met* $\sqsubseteq$ *een poset zijn.* P *is een oneindig stijgende (dalende) verzameling desda voor iedere* x *in* P *de verzameling van bovengrenzen (ondergrenzen) in* P *oneindig is.*

Een open interval op de verzameling reële getallen met de ordening *groter dan of gelijk aan* is zowel oneindig stijgend als dalend.

Definitie 79 *Cover Laat P met $\sqsubseteq$ een poset zijn, x en y twee elementen in P en x een strikte bovengrens van y. x is een cover van y desda voor alle strikte bovengrenzen z van y geldt dat $z \sqsubseteq x$ of dat x en z onvergelijkbaar zijn.*

Vier is een cover van drie op de verzameling natuurlijke getallen met de ordening *groter dan of gelijk aan*. Een reëel getal heeft op dezelfde ordening geen covers want tussen twee reële getallen ligt altijd een ander reëel getal.

Definitie 80 *Interval Laat P met $\sqsubseteq$ een poset zijn. Een interval $S = [a, b]$ is de verzameling $\{x \epsilon P \mid a \sqsubseteq x \wedge x \sqsubseteq b\}$.*

Definitie 81 *Keten Laat P met $\sqsubseteq$ een poset zijn en S een deelverzameling van P. S is een keten desda S met $\sqsubseteq$ totaal geordend is.*

Definitie 82 *Totale keten Laat P met $\sqsubseteq$ een poset zijn en S een keten van P. S is een totale keten van P desda S niet met een element van P kan worden uitgebreid dat groter is dan een element van S en kleiner dan een element van S zdd S met dit element weer een keten is.*

Definitie 83 *Duale ordening Laat P met $\sqsubseteq$ een poset zijn. $\sqsupseteq$, de duale ordening, is de ordening gedefinieerd door $x \sqsupseteq y$ desda $y \sqsubseteq x$. P met $\sqsupseteq$ is ook een poset.*

4.3.2 Kwasi-ordening

Een kwasi-ordening is zwakker dan een partiële ordening; de betrekking hoeft niet aan de symmetrie-eis te voldoen.

Definitie 84 *Kwasi-Ordening Laat P een verzameling met een betrekking $\sqsubseteq$ zijn. $\sqsubseteq$ is een kwasi-ordening desda voor alle x, y en z in P geldt:*

1. *$x \sqsubseteq x$ (de betrekking is reflexief);*
2. *Als $x \sqsubseteq y$ en $y \sqsubseteq z$ dan volgt $x \sqsubseteq z$ (de betrekking is transitief);*

Definitie 85 *Ekwivalentieklasse Laat P een verzameling met kwasi-ordening $\sqsubseteq$ zijn en x een element van P. De ekwivalentieklasse van x, $[x]$, is gelijk aan de verzameling $\{y \mid x \sqsubseteq y \wedge y \sqsubseteq x\}$.*

De definities op ordeningen gelden ook op kwasi-ordeningen. Een kwasi-ordening induceert een natuurlijke partiële ordening op de ekwivalentieklassen, waarbij $[x] \sqsubseteq [y]$ desda er een x' en y' zijn zdd $x' \epsilon [x]$, $y' \epsilon [y]$ en $x' \sqsubseteq y'$.[1]

[1] Strikt genomen is de ordening op de ekwivalentieklassen een andere ordening dan de kwasi-ordening op de elementen en moet er een ander symbool voor deze (geïnduceerde) ordening worden gehanteerd. Als het onderscheid tussen beide ordeningen duidelijk is wordt meestal hetzelfde symbool gehanteerd.

4.3.3 Tralie

Een tralie is een poset met 'mooie' eigenschappen.

Definitie 86 *Tralie Laat P met $\sqsubseteq$ een poset zijn. P met $\sqsubseteq$ is een tralie desda voor alle x en y in P zowel een supremum als infimum van $\{x, y\}$ bestaat.*

Definitie 87 *Modulair Laat P met $\sqsubseteq$ een tralie zijn. P met $\sqsubseteq$ is modulair desda voor alle x, y, en z in P geldt:*

$$(x \sqsubseteq y) \Rightarrow (x \sqcup (y \sqcap z)) = ((x \sqcup y) \sqcap z) \tag{4.1}$$

Definitie 88 *Kompleet Laat P met $\sqsubseteq$ een tralie zijn. P is kompleet desda voor iedere deelverzameling van P een supremum en infimum bestaat.*

4.4 Generalisatiemodellen

4.4.1 Regels

De in de vorige paragraaf geïntroduceerde ordening op formules kan ook opgevat worden als een ordening op regels [30], in het bijzonder een ordening op formules binnen de Horn-clause logika (als gevolg van de procedurele semantiek). Het generalisatiemodel kan dan worden gebruikt als basis om de zoekruimte van regels te organiseren:

> Regardless of the representation language used, a key part of the induction process, as it is envisaged by many researchers in the AI discipline, is a search through a space of rules. A model of generalization provides a basis for organizing this search space. Briefly, rule R_1 is more general than rule R_2, or R_2 is more specific than R_1, if in any world R_1 can be used to show at least the same results as R_2 (adapted from Mitchell [30]). Rules can be organized into a structure called a hierarchy. All clauses below a node in a *specialization (generalisation) hierarchy* are specialisations (generalizations) of the node, and, furthermore, the hierarchy contains all rules more specific (general) than the root. – Buntine [9]

Het gebruik van een generalisatie- (of specialisatie-)hierarchie heeft drie voordelen [9]:

1. Takken van de zoekruimte kunnen worden gesnoeid als alle generalisaties een bepaalde niet-toegestane eigenschap hebben;
2. Kennis van de generalisatie van een gewenste regel kan worden gebruikt om de zoekruimte te beperken. Een bijkomend voordeel is dat het testen op konsistentie van een regel met de beschikbare gegevens eenvoudiger wordt.
3. Als twee regels generalisaties van een onbekende regel zijn, is het supremum van deze regels dit misschien ook.

4.4.2 Zwakkere modellen

Het in dit hoofdstuk besproken generalisatiemodel is gebaseerd op logische implikatie. Implementatie van dit model is echter problematisch. Daarom wordt in de praktijk vaak gebruik gemaakt van zwakkere modellen, modellen die het aantal generalisaties beperken. Een voorbeeld van een zwakker generalisatiemodel is het model van Plotkin [47], gebaseerd op *subsumptie*.

Een andere reden om zwakkere modellen te analyseren is dat zij een beter inzicht geven in het gebruikte model. Bovendien kunnen de suprema en covers van zwakkere modellen soms dienen als suprema en covers van het gehanteerde model.

In het volgende hoofdstuk worden vijf modellen besproken van generalisaties van clausal forms zonder achtergrondtheorie, waarbij ieder model sterker is dan zijn voorganger. Het sterkste model is gebaseerd op logische implikatie. De ekwivalentieklasse van een sterker model is gelijk aan de vereniging van een aantal ekwivalentieklassen van een zwakker model.

Vervolgens worden in het daarop volgende hoofdstuk generalisaties van clausal forms met een achtergrondtheorie geanalyseerd. *Subsumptie* [47] wordt uitgebreid tot *relatieve subsumptie* [48] en *gegeneraliseerde subsumptie* [9] en het resolutie-algoritme kan worden geïnverteerd [31].

Hoofdstuk 5

Generalisatie zonder achtergrondtheorie

5.1 Inleiding

De verzameling wff van de clause logika met de kwasi-ordening *algemener dan* volgens logische implikatie induceert een tralie met als supremum van twee wff de konjunktie en als infimum de disjunktie ervan.

Bij induktief leren wordt niet een theorie gegeneraliseerd maar clausal forms van die theorie. In dit hoofdstuk worden de eigenschappen van de verzameling clausal forms met de kwasi-ordening *algemener dan* geanalyseerd. De induktieve operatoren die geen gebruik maken van achtergrondkennis worden puur induktieve operatoren genoemd [55]. De meeste resultaten zijn ook gepubliceerd in [44].

Op de verzameling clausal forms worden vijf verschillende kwasi-ordeningen gedefinieerd waarbij iedere kwasi-ordening sterker is dan zijn voorganger; als een clausal form in een bepaalde kwasi-ordening algemener is dan een andere clausal form, is hij dit ook in een sterkere kwasi-ordening. De vijfde en sterkste kwasi-ordening is *algemener dan* volgens logische implikatie.

De verschillende kwasi-ordeningen induceren verschillende ekwivalentieklassen. De ekwivalentieklasse van een sterkere kwasi-ordening is de vereniging van een aantal ekwivalentieklassen van een zwakkere kwasi-ordening.

De duale kwasi-ordeningen zijn specialisaties en zijn van belang voor refinement-operatoren, bijvoorbeeld in het model van Shapiro.

De reden dat de verschillende kwasi-ordeningen zijn gedefinieerd is dat zij kunnen worden vergeleken. De relaties van covers en suprema van de verschillende kwasi-ordeningen zijn van belang bij het vinden van covers en suprema van sterkere kwasi-ordeningen, in het bijzonder subsumptie en logische implikatie.

Het logische supremum van een verzameling wff – de konjunktie ervan – is logisch ekwivalent met deze verzameling, het supremum van een verzameling clausal forms is bijna altijd een generalisatie van deze verzameling (voor iedere kwasi-ordening).

Als het supremum van een aantal clausal forms bestaat is het van belang dat dit supremum konsistent is. Het supremum is een specialisatie van alle generalisaties van de clausal forms: als de clausal forms een generalisatie hebben, dan is hun supremum geldig [35].

Als het supremum niet bestaat doordat de verzameling strikte bovengrenzen van een verzameling clausal forms meerdere minimale elementen heeft, dan hoeft slechts één van deze minimale elementen konsistent te zijn. Als er geen minimale elementen zijn (door oneindige ketens) dan is het niet bekend hoever kan worden gegeneraliseerd.

5.1.1 Kwasi-ordeningen

Vijf kwasi-ordeningen worden in dit hoofdstuk besproken: substitutie-generalisatie[1], permutatie-generalisatie, set-generalisatie, subsumptie-generalisatie en logische generalisatie.

Clausal form C_1 is een substitutie-generalisatie van clausal form C_2 desda C_2 (deduktief) van C_1 kan worden afgeleid met behulp van het substitutie-axioma, ofwel desda er een substitutie θ is zdd $C_2 = C_1\theta$.

Een ekwivalentieklasse onder de substitutie-kwasi-ordening is gelijk aan alle alfabetische varianten van een clausal form.

Als de verzameling clausal forms wordt uitgebreid met de lege theorie induceert deze kwasi-ordening een tralie. De deelverzameling van bovengrenzen van een clausal form (inklusief de clausal form zelf) met deze kwasi-ordening induceert ook een tralie.

Clausal form C_1 is een permutatie-generalisatie van clausal form C_2 als C_2 van C_1 kan worden afgeleid met behulp van het substitutie-axioma en een permutatie van literals.

Er wordt aangetoond dat een substitutie-cover ook een permutatie-cover is en dat een permutatie-cover een substitutie-cover van een permutatie-ekwivalente clausal form is. Het gevolg is dat het vinden van alle permutatie-covers zich kan beperken tot het vinden van alle substitutie-covers.

Er wordt bewezen dat een permutatie-supremum niet hoeft te bestaan. De permutatie-kwasi-ordening induceert dus geen tralie.

Clausal form C_1 is een set-generalisatie van clausal form C_2 als C_2 van C_1 kan worden afgeleid met behulp van het substitutie-axioma, permutaties van literals en het herhalen van literals of het verwijderen van een literal die meerdere keren voorkomt.

De ekwivalentieklasse onder de set-kwasi-ordening is gelijk aan de alfabetische variaties van een clause. In deze (en een sterkere) kwasi-ordening wordt daarom

[1] Nienhuys-Cheng [44] noemt dit syntaktische generalisatie, maar daar alle generalisaties in feite syntaktische generalisaties zijn en om verwarring te voorkomen heb ik deze, meer voor de hand liggende, naam gekozen.

gesproken over clauses.

Een strikte substitutie-generalisatie is ook een strikte set-generalisatie. Een substitutie-cover van een clausal form hoeft echter geen set-cover te zijn.

Evenals het permutatie-supremum hoeft een set-supremum niet te bestaan.

Clausal form C_1 is een subsumptie-generalisatie van clausal form C_2 als C_2 van C_1 kan worden afgeleid met behulp van het substitutie-axioma, permutaties van literals, het verwijderen van een literal die meerdere keren voorkomt en het toevoegen van een willekeurige literal (via de stelling $\alpha \rightarrow \alpha \vee \beta$).

Een strikte substitutie-generalisatie hoeft geen strikte subsumptie-generalisatie te zijn. Plotkin [47] heeft aangetoond dat het supremum van een verzameling clauses bestaat en noemde het de *least general generalisation.* De subsumptie-kwasi-ordening induceert een tralie.

Clausal form C_1 is een logische generalisatie van clausal form C_2 desda C_2 een logisch gevolg van C_1 is, ofwel afleidbaar is van C_1.

5.2 Substitutie

In deze paragraaf worden de eigenschappen van substitutie-generalisatie voor clausal forms geanalyseerd. Reynolds [51] introduceerde deze eigenschappen voor atomen en Plotkin [47] werkte het uit voor woorden.

Nienhuys-Cheng [44] laat zien dat het ook voor clausal forms geldt, daar een clausal form als gegeneraliseerd woord kan worden opgevat. Een clausal form kan in prefix-notatie worden geschreven, de notatie van een clausal form wordt dan:

(5.1) $\vee(L_1, \ldots, L_n)$

Definitie 89 $\geq_s$ *Laat C_1 en C_2 een clausal form of $\blacksquare$ zijn. $C_1 \geq_s C_2$[2], C_1 is een substitutie-generalisatie van C_2, desda één van de volgende drie gevallen optreedt:*

- $C_1 = \square$;
- $C_2 = \blacksquare$;
- *Er is een substitutie θ zdd $C_2 = C_1\theta$.*

Definitie 90 $\sim_s$ *Laat C_1 en C_2 een clausal form of $\blacksquare$ zijn. $C_1 \sim_s C_2$[3], C_1 is substitutie-ekwivalent aan C_2, desda $C_1 \geq_s C_2$ en $C_2 \geq_s C_1$.*

Stelling 12 *Ekwivalentie Laat C_1 en C_2 een clausal form of $\blacksquare$ zijn. $C_1 \sim_s C_2$ desda C_1 en C_2 alfabetische variaties zijn.*

Bewijs 6 *Dit is een generalisatie van deze eigenschap van atomen [51].*

[2] Reynolds gebruikt eveneens $\geq$, Plotkin gebruikt echter $\leq$.

[3] Reynolds gebruikt $\simeq$.

Definitie 91 $>_s$ *Laat* C_1 *en* C_2 *een clausal form of* ■ *zijn.* $C_1 >_s C_2$, C_1 *is een strikte substitutie-generalisatie van* C_2, *desda* $C_1 \geq_s C_2$ *en* $C_2 \not\geq_s C_1$.

Definitie 92 $\rightarrow_s$ *Laat* C_1 *en* C_2 *een clausal form of* ■ *zijn.* $C_1 \rightarrow_s C_2$, C_1 *is een substitutie-cover van* C_2, *desda* $C_1 >_s C_2$ *en er geen clausal form* C_3 *is zdd* $C_1 >_s C_3 >_s C_2$.

Stelling 13 *Cover Laat* C_1 *en* C_2 *een clausal form of* ■ *zijn.* C_1 *is een substitutie-cover van* C_2 *desda één van volgende waar is:*

1. C_1 *en* C_2 *zijn clausal forms zdd voor een konstante* a *geldt dat* $C_2 \sim_s C_1\{x/a\}$, *waarbij* x *een variabele is die voorkomt in* C_1.

   Een substitutie-cover van type 1 kan worden gevonden door het vervangen van een aantal voorkomens van een konstante door een nieuwe variabele.

2. C_1 *en* C_2 *zijn clausal forms zdd* $C_2 \sim C_1\{x/y\}$, *waarbij* x *en* y *verschillende variabelen zijn die voorkomen in* C_1.

   Een substitutie-cover van type 2 kan worden gevonden door het vervangen van een aantal (maar niet alle) voorkomens van een variabele door een nieuwe variabele.

3. C_1 *en* C_2 *zijn clausal forms zdd voor een k-plaatsige funktie* f *geldt* $C_2 \sim_s C_1\{x/f(z_1, \ldots, z_k)\}$, *waarbij* x *een variabele is die voorkomt in* C_1 *en* z_i *verschillende variabelen zijn die niet voorkomen in* C_1.

   Een substitutie-cover van type 3 kan worden gevonden door het vervangen van alle voorkomens van een term door een nieuwe variabele, waarbij die term een funktie is van verschillende variabelen die alleen in die term in de clausal form voorkomen.

4. $C_1 = \square$ *en* C_2 *is een clausal form waarin alle termen verschillende variabelen zijn.*

5. C_1 *is een grond clausal form en* $C_2 =$ ■.

Bewijs 7 *Dit is een generalisatie van deze eigenschap van atomen [51].*

Laat $C = P(x, x, f(y)) \vee Q(f(y), a)$, dan zijn de covers van C alfabetische variaties van:

- $P(z, x, f(y)) \vee Q(f(y), a)$
- $P(x, x, f(z)) \vee Q(f(y), a)$
- $P(x, x, z) \vee Q(z, a)$
- $P(x, x, f(y)) \vee Q(f(y), z)$

5.2.1 Inverse substitutie

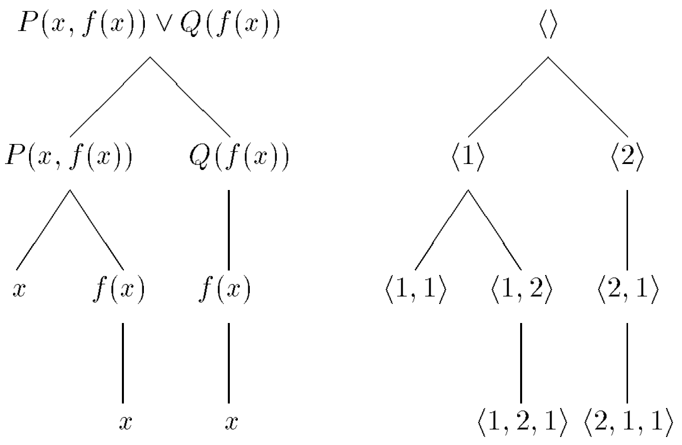
Bij deduktieafleidingen worden substituties van clausal forms toegestaan via het substitutie-axioma, bij induktie wordt deze bewerking geïnverteerd.

Definitie 93 *Inverse substitutie Laat C_1 en C_2 twee clausal forms zijn en θ een substitutie zdd $C_1 = C_2\theta$. θ^{-1}, de inverse substitutie van θ, is gelijk aan een transformatie zdd $C_1\theta^{-1} = C_2$, ofwel zdd:*

(5.2) $C_1\theta\theta^{-1} = C_1$

Een substitutie vervangt variabelen door termen. Bij inverse substitutie wordt deze bewerking omgekeerd: termen worden door variabelen vervangen. Een substitutie vervangt alle variabelen door dezelfde term, een inverse substitutie vervangt een aantal identieke termen door een nieuwe variabele. Substituties zijn dus onafhankelijk van de wff, inverse substituties niet. Bij inverse substituties zijn ook de posities van de termen binnen de clausal form van belang.

Een clausal form kan worden gerepresenteerd met een boom. In figuur 5.1 is de boomrepresentatie van $P(x, f(x)) \vee Q(f(x))$ weergegeven[4]. Een term van de clausal form heeft een vaste positie in de boom.



Figuur 5.1: Boomstruktuur van $P(x, f(x)) \vee Q(f(x))$

Definitie 94 *Positie Een positie is een reeks $\langle p_1, p_2, \ldots, p_n\rangle$ positieve natuurlijke getallen.*

De positie van een literal of term in een clausal form, literal of term wordt induktief gedefinieerd. Laat E een clausal form, literal of term zijn.

[4] Informatici hebben de vreemde gewoonte de wortel van de boom altijd bovenaan weer te geven en de bladeren onderaan.

- $\langle\rangle$ *is de positie van* E *in* E.
- *Als* $E' = Y(E_1, E_2, \ldots, E_k)$ *een clausal form, literal of term is met positie* $\langle p_1, p_2, \ldots, p_n\rangle$ *in* E, *dan is* $\langle p_1, p_2, \ldots, p_n, i\rangle$ *de positie van* E_i *in* E.

Als E' *een literal (term) is met positie* p *in* E, *dan is de kombinatie* (E', p) *een literal- (term-)voorkomen van* E.

De posities van de voorkomens van $f(x)$ in $P(x, f(x)) \vee Q(f(x))$ zijn gelijk aan $\langle 1, 2\rangle$ en $\langle 2, 1\rangle$ en $(f(x), \langle 1, 2\rangle)$ en $(f(x), \langle 2, 1\rangle)$ zijn termvoorkomens.

Posities kunnen met elkaar worden vergeleken. Een positie is een subreeks van een andere positie desda hij een voorouder in de boom is.

Definitie 95 *Subreeks Laat* $p_1 = \langle p_{1,1}, p_{1,2}, \ldots, p_{1,n}\rangle$ *en* $p_2 = \langle p_{2,1}, p_{2,2}, \ldots, p_{2,m}\rangle$ *twee posities zijn.* p_1 *is een subreeks van* p_2 *desda* $n \leq m$ *en* $p_{1,i} = p_{2,i}$ $(1 \leq i \leq n)$.

Als $v_1 = (t_1, p_1)$ en $v_2 = (t_2, p_2)$ twee termvoorkomens van dezelfde clausal form zijn en p_1 een subreeks van p_2 is, dan is v_2 een subtermvoorkomen van v_1. Deze relatie wordt genoteerd met $v_1 \geq v_2$.

Als een termvoorkomen een subtermvoorkomen van een ander termvoorkomen is, kan ook de relatieve positie worden bepaald. De relatieve positie wordt genoteerd met $p_2 - p_1 = \langle p_{2,n+1}, p_{2,n+2}, \ldots, p_{2,m}\rangle$.

Een inverse substitutie is gelijk aan het vervangen van identieke termen in termvoorkomens door nieuwe variabelen en wordt genoteerd met:

(5.3) $\{(t_1, p_1)/x_1, \ldots, (t_1, p_k)/x_1, \ldots, (t_n, p_m)/x_n\}$

5.2.2 Termpartitie

Een inverse substitutie kan ook worden gerepresenteerd met behulp van een partitie van termvoorkomens [37]. In een partitie zitten die termvoorkomens die door dezelfde variabele worden vervangen.

Alvorens termpartities kunnen worden gedefinieerd, moeten eerst de begrippen minimaal en toegestaan worden geïntroduceerd. Een verzameling termvoorkomens is minimaal desda er geen twee termvoorkomens zijn waarbij de ene een voorouder in de boom is van de ander. Een verzameling termvoorkomens is toegestaan als de verzameling minimaal is en alle variabelen van de clausal form een subtermvoorkomen van een termvoorkomen zijn.

Definitie 96 *Minimaal Laat* C *een clausal form zijn,* $T(C)$ *de verzameling van alle termvoorkomens van* C *en* T *een deelverzameling van* $T(C)$. T *is minimaal desda er van alle termvoorkomens in* T *geen subtermvoorkomens in* T *voorkomen.*

Definitie 97 *Toegestaan Laat* C *een clausal form zijn,* $T(C)$ *de verzameling van alle termvoorkomens van* C *en* T *een minimale deelverzameling van* $T(C)$. T *is toegestaan desda er voor alle variabelenvoorkomens* (x, p) *in* $T(C)$ *een* (t, p') *in* T *is zdd* $(t, p') \geq (x, p)$.

Definitie 98 *Termpartitie Laat C een clausal form zijn, $T(C)$ de verzameling van alle termvoorkomens van C en T een toegestane deelverzameling van $T(C)$. Een termpartitie van T is een verzameling van niet-lege deelverzamelingen $\{B_1, B_2, \ldots, B_n\}$ van T zdd:*

- *$B_1 \cup B_2 \cup \ldots \cup B_n = T$;*
- *$B_i \cap B_j = \emptyset$ als $i \neq j$;*
- *Ieder blok B_i bevat alleen voorkomens van dezelfde term.*

Laat $C = P(x, f(x)) \vee Q(f(x))$. De substitutie-generalisatie $C' = P(x, y) \vee Q(y)$ kan worden weergegeven met:

- een inverse substitutie: $\{(f(x), \langle 1, 2\rangle)/y, (f(x), \langle 2, 1\rangle)/y\}$;
- een termpartitie: $\{\{(f(x), \langle 1, 2\rangle), (f(x), \langle 2, 1\rangle)\}, \{(x, \langle 1, 1\rangle)\}\}$.

Termpartities vergelijken

De termpartities kunnen worden gebruikt om twee generalisaties van een clausal form met elkaar te vergelijken. Een termpartitie is hoger desda de gerepresenteerde clausal form algemener is [38].

Definitie 99 *Hoger dan Laat C een clause zijn, $T(C)$ de verzameling van alle termvoorkomens, T_1 en T_2 twee deelverzamelingen van $T(C)$ en Ω_1 en Ω_2 termpartities van T_1 en T_2. $\Omega_1 \geq \Omega_2$, Ω_1 is hoger dan Ω_2, desda aan de volgende twee voorwaarden voldaan is:*

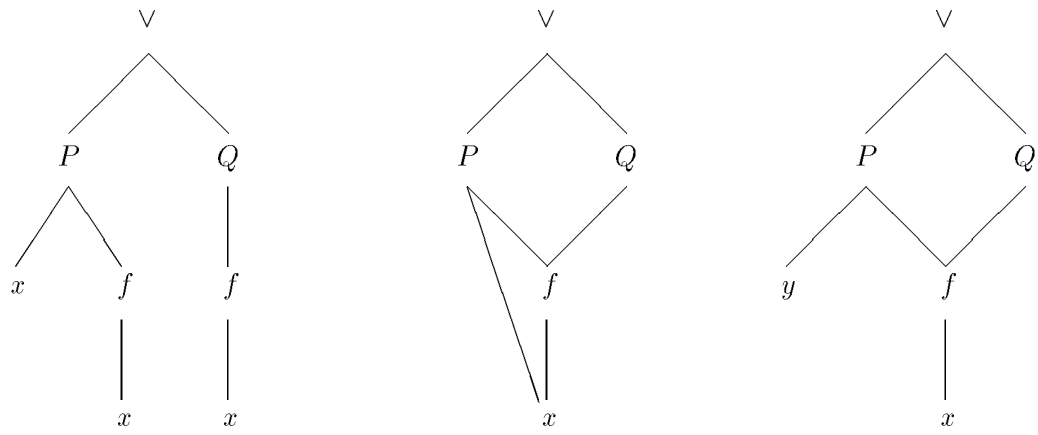
1. *Voor iedere (t_2, p_2) in T_2 is er een (t_1, p_1) in T_1 zdd $(t_1, p_1) \geq (t_2, p_2)$;*
2. *Laat (t_1, p_1) in een blok $B_1 = \{(t_1, p_{1,1}), \ldots, (t_1, p_{1,n})\}$ van Ω_1 zijn en (t_2, p_2) in een blok $B_2 = \{(t_2, p_{2,1}), \ldots, (t_2, p_{2,m})\}$ van Ω_2. Als $(t_1, p_1) \geq (t_2, p_2)$ dan is $m \geq n$ en is er een permutatie π van $\{1, \ldots, m\}$ zdd $p_{2,\pi(i)} - p_{1,i} = p_2 - p_1$ $(1 \leq i \leq n)$.*

Covers

De covers van een clausal form kunnen met behulp van termpartities worden gerepresenteerd. Allereerst wordt de triviale partitie gedefinieerd, die korrespondeert met ekwivalente clausal forms.

Definitie 100 *Triviale partitie Laat C een clausal form zijn en T een termpartitie van C. T is triviaal als er alleen variabelenvoorkomens in voorkomen en er geen twee blokken met dezelfde variabele in voorkomen.*

Stelling 14 *Covers Laat C een clausal form zijn met triviale termpartitie Ω. De covers van C zijn (afgezien van de lege clause):*



termvoorkomens termen een generalisatie

Figuur 5.2: Boomstrukturen

- *Type 1. Laat a een konstante in C zijn en $B = \{(a, p_1), \ldots, (a, p_n)\}$ een niet-lege deelverzameling van alle a-voorkomens in C. De partitie $\Omega \cup \{B\}$ representeert een substitutie-cover van type 1 van C.*

- *Type 2. Laat x een variabele in C zijn, $B = \{(x, p_1), \ldots, (x, p_n)\}$ de verzameling van alle x-voorkomens in C en B_1 en B_2 twee niet-lege deelverzamelingen van B zdd $B_1 \cup B_2 = B$ en $B_1 \cap B_2 = \emptyset$. De partitie $(\Omega - \{B\}) \cup \{B_1, B_2\}$ representeert een substitutie-cover van type 2 van C.*

- *Type 3. Laat $t = f(x_1, x_2, \ldots, x_n)$ een term in C zijn, f een funktiesymbool, x_i een aantal verschillende variabelen die alleen in een t voorkomen in C, $B = \{(t, p_1), \ldots, (t, p_n)\}$ de verzameling van alle t-voorkomens in C en $B_i = \{(x_i, p_1), \ldots, (x_i, p_n)\}$ de verzameling van alle x_i-voorkomens in C. De partitie $(\Omega - \{B_1, \ldots, B_n\}) \cup \{B\}$ representeert een substitutie-cover van type 3 van C.*

Termvoorkomens of termpartities

Een clausal form kan worden gerepresenteerd met een verzameling termvoorkomens, maar ook met een verzameling van termen (met posities). In figuur 5.2 staan de twee verschillende representaties als boomstrukturen van de clausal form $P(x, f(x)) \vee Q(x)$ (vergelijk figuur 5.1). De eerste struktuur wordt termvoorkomenstruktuur en de tweede termstruktuur genoemd. Merk op dat niet iedere representatie van een termstruktuur grafisch mooi is weer te geven, bij een implementatie is dit uiteraard niet van belang.

De meeste implementaties van Prolog hanteren de eerste struktuur als representatiemechanisme, bijvoorbeeld de Prolog-V implementatie in Smalltalk-V.

Iedere knoop in de boom wordt gerepresenteerd met een objekt. Als alternatief zou iedere term met een objekt kunnen worden gerepresenteerd.

Het nadeel is dat een clausal form eerst moet worden herschreven naar deze termstruktuur. Een voordeel is dat de representatie efficiënter gebruik maakt van het geheugen. Het voordeel bij substituties is dat een variabele slecht op één plaats hoeft te worden gesubstitueerd. Bij een substitutie blijft de boom immers intakt, hij wordt slechts uitgebreid.

Het belangrijkste voordeel is dat de boom ook bottom-up in plaats van top-down doorlopen kan worden. In plaats van te beginnen met de root, $\vee$, en dan door de boom te wandelen tot de bladeren zijn bereikt, kunnen de bladeren ook direkt worden benaderd. Substituties zijn efficiënter te implementeren, al zullen unifikaties van twee variabelen wel nauwkeurig moeten worden afgehandeld[5].

In het bijzonder zijn generalisaties snel te konstrueren, namelijk door termpartities te genereren. In figuur 5.2 is te zien hoe de generalisatie $P(y, f(x)) \vee Q(x)$ wordt gekonstrueerd door de term x te splitsen in partities van x en y.

Verschillende generalisaties van dezelfde clausal form kunnen efficiënt worden vergeleken door gebruik te maken van *hoger dan*, zie definitie 99.

5.2.3 Supremum en infimum

Het berekenen van infima en suprema in de natuurlijke partiële ordening is eenvoudig, zoals Reynolds [51] heeft aangetoond.

Stelling 15 *Infimum Laat C_1 en C_2 een clausal form of $\blacksquare$ zijn. Het infimum van C_1 en C_2 kan worden berekend met de volgende algoritme:*

- *Als $C_1 = \square$ dan is $C_1 \sqcap C_2 = C_2$;*
- *Als $C_2 = \square$ dan is $C_1 \sqcap C_2 = C_1$;*
- *Als $C_1 = \blacksquare$, $C_2 = \blacksquare$ of C_1 en C_2 niet unificeerbaar zijn, dan is $C_1 \sqcap C_2 = \blacksquare$;*
- *Anders is $C_1 \sqcap C_2$ de (meest algemene) unifikatie van C_1 en C_2.*

Bewijs 8 *Zie [51, Theorem 1].*

Stelling 16 *Sup-vergelijkbaar Laat $C_1 = L_{1,1} \vee \ldots \vee L_{1,n}$ en $C_2 = L_{2,1} \vee \ldots L_{2,n}$ een clausal form of $\blacksquare$ zijn. C_1 en C_2 zijn sup-vergelijkbaar desda één van de volgende twee gevallen optreedt:*

- *C_1 of C_2 is gelijk aan $\square$ of $\blacksquare$;*
- *$L_{1,i}$ en $L_{2,i}$ $(1 \leq i \leq n)$ hebben hetzelfde teken en beginnen met hetzelfde predikaatsymbool.*

[5] Ik stel mij voor dat bij iedere clausal form een aantal vaste objekten wordt gedefinieerd die variabelen representeren (onafhankelijk van het feit of ze in de clausal form voorkomen). Een bijkomend voordeel is dat clausal forms automatisch apart gestandaardiseerd zijn.

Als twee clausal forms unificeerbaar zijn, zijn zij ook sup-vergelijkbaar. Als zij echter sup-vergelijkbaar zijn, hoeven zij nog niet unificeerbaar te zijn.

Stelling 17 *Supremum Laat C_1 en C_2 twee clausal forms zijn. Het supremum van C_1 en C_2 kan worden gevonden met de volgende algoritme van Reynolds [51]:*

1. *Als $C_1 = \Box$ of $C_2 = \Box$ of C_1 en C_2 niet sup-vergelijkbaar zijn, dan is $C_1 \sqcup C_2 := \Box$;*
2. *Als $C_1 = \blacksquare$ dan is $C_1 \sqcup C_2 := C_2$;*
3. *Als $C_2 = \blacksquare$ dan is $C_1 \sqcup C_2 := C_1$;*
4. *Anders berekent de volgende anti-unifikatie algoritme[6] $C_1 \sqcup C_2$. Het werkt systematisch alle termen af en indien zij verschillen wordt de term vervangen door een variabele die reeds identieke termen vervangen heeft (stap d) of door een nieuwe variabele (stap e)[7]:*
   (a) *$\overline{C}_1 := C_1$, $\overline{C}_2 := C_2$, $\zeta := \{\}$, $\eta := \{\}$, $i := 0$, $\{z_i\} :=$ een rij variabelen zijn die niet voorkomen in C_1 of C_2;*
   (b) *Als $\overline{C}_1 = \overline{C}_2$, stop dan met $C_1 \sqcup C_2 := \overline{C}_1$;*
   (c) *Laat k de positie zijn van het eerste symbool waarin $\overline{C}_1$ en $\overline{C}_2$ verschillen, en laat t_1 en t_2 de termen zijn die beginnen op de k-de positie in respektievelijk $\overline{C}_1$ en $\overline{C}_2$;*
   (d) *Als er een j is zdd $1 \leq j \leq i$, $z_j\zeta = t_1$ en $z_j\eta = t_2$, vervang dan de termen op de k-de positie van $\overline{C}_1$ en $\overline{C}_2$ door z_j en ga naar b;*
   (e) *Anders $i := i+1$; vervang de termen op de k-de positie van $\overline{C}_1$ en $\overline{C}_2$ door z_i; $\zeta := \zeta \cup \{z_i/t_1\}$ en $\eta := \eta \cup \{z_i/t_2\}$. Ga naar b.*

Een voorbeeld van het supremum van twee clausal forms:

- $C_1 = P(x, a) \vee Q(x, f(f(a)))$;
- $C_2 = P(b, y) \vee Q(b, f(g(a)))$;
- $C_1 \sqcup C_2 = P(x, y) \vee Q(x, f(z))$.

Stelling 18 *Supremum Laat S een (mogelijk oneindige) verzameling van clausal forms zijn. Het supremum van S bestaat.*

Het supremum van S is gelijk aan $\Box$ desda $\Box \epsilon S$ of er twee clausal forms in S zijn die niet sup-vergelijkbaar zijn.

Bewijs 9 *Dit is een generalisatie van deze eigenschap voor woorden [47].*

[6] Dezelfde algoritme werd gelijktijdig onafhankelijk door Plotkin gevonden.

[7] Bij unifikatie worden de variabelen (indien mogelijk) juist door termen vervangen; in die zin is deze algoritme dus de inverse van unifikatie.

Termvoorkomens of termpartities

Bij het konstrueren van infima en suprema kan, net als bij het berekenen van onder- en bovengrenzen, worden geanalyseerd of een representatie als termvoorkomenstruktuur of als termstruktuur efficiënter is.

Bij het berekenen van infima en suprema worden de clausal forms altijd top-down doorlopen, zoals uit de bovenstaande algoritme blijkt. Bij unifikatie is er bij een representatie als termstruktuur het voordeel dat sneller kan worden bepaald of twee clausal forms unificeerbaar zijn: als er een variabele wordt vervangen, gebeurt dat direkt voor alle voorkomens. Bij suprema geldt dit voordeel niet; dezelfde variabele of term kan door verschillende variabelen worden vervangen.

Als twee clausal forms gerepresenteerd zijn met termpartities van dezelfde clausal form kunnen zij wel bottom-up worden doorlopen, wat een voordeel van een termstruktuur zou kunnen zijn. In de praktijk zal dit echter slechts weinig voorkomen.

5.2.4 Tralie

Daar het supremum en infimum altijd bestaan, is de natuurlijke partiële ordening van de substitutie-kwasi-ordening een tralie.

- Ekwivalentieklasse: $[C] = \{C_i \mid C_i \sim_s C\}$.
- Ordening: $[C_1] \geq_s [C_2]$ desda $C_1 \geq_s C_2$.
- Supremum: $[C_1] \sqcup [C_2] = [C_1 \sqcup C_2]$.
- Infimum: $[C_1] \sqcap [C_2] = [C_1 \sqcap C_2]$.

De tralie is kompleet [51] daar iedere (mogelijk oneindige) verzameling altijd een infimum en supremum heeft. De tralie is niet modulair, zoals blijkt uit het voorbeeld van Reynolds [51, p. 141]: van $P(f(x), f(x))$ naar $P(x, y)$ lopen o.a. totale ketens via $P(x, x)$ en via $P(f(x), f(y))$ en $P(f(x), y)$.

Bovengrenzen clausal form

Op de verzameling van bovengrenzen van een clausal form (zonder de lege clause) induceert de substitutie-kwasi-ordening een tralie met als minimum de clausal form zelf en als maximum de clausal form waarin alle termen vervangen zijn door verschillende variabelen.

Keten

Stelling 19 *Keten Laat C_1 en C_2 twee clausal forms zijn. Als $C_1 >_s C_2$ dan is er een totale keten met maximum C_1 en minimum C_2.*

Bewijs 10 *Dit is een generalisatie van deze eigenschap voor atomen [51].*

De kompleksiteitsmaat van een clausal form kan worden gegeven met behulp van *size* [51].

Definitie 101 *Size Laat C is een clausal form of $\blacksquare$ zijn. $Size(C)$ is gelijk aan:*

1. $size(\square) = 0$;
2. $size(\blacksquare) = \infty$;
3. $size(C)$ = *het aantal symboolvoorkomens – het aantal verschillende variabelen in C.*

   Een symbool is een predikaatsymbool, een variabele, een konstante of een funktiesymbool.

Het is eenvoudig in te zien dat geldt:

- Als $C_1 \sim_s C_2$ dan is $size(C_1) = size(C_2)$.
- Als $C_1 >_s C_2$ dan is $size(C_1) < size(C_2)$.

Twee clausal forms met dezelfde size zijn ekwivalent of onvergelijkbaar en een clausal form met een kleinere size is algemener of onvergelijkbaar.

Laat C_1 en C_2 twee clausal forms zijn en C_1 een cover van C_2. Dan geldt:

- Type 1: $size(C_2) - size(C_1) = 1$;
- Type 2: $size(C_2) - size(C_1) = 1$;
- Type 3: $size(C_2) - size(C_1) = (n-1)*k+1$, waarbij n het aantal vervangen k-plaatsige funkties is.

Het verschil in size van maximum en minimum (plus één) in een keten is een bovengrens van het aantal elementen in de keten. Er zijn dus geen oneindige stijgende ketens, want voor iedere clausal form is het aantal bovengrenzen eindig.

Er zijn wel oneindige dalende ketens. Een voorbeeld van een oneindige dalende keten is $\{P(x), P(f(x)), P(f(f(x))), \ldots\}$.

5.3 Permutatie

Een permutatie van een clausal form ontstaat door een aantal literals van de clausal form te verwisselen. Permutaties van een clausal form zijn ekwivalent.

Definitie 102 *Permutatie Laat $C = L_1 \vee \ldots \vee L_n$ een clausal form zijn en π een permutatie van $\{1, \ldots, n\}$. De permutatie $C\pi$ van C is de clausal form $L_{\pi(1)} \vee \ldots \vee L_{\pi(n)}$.*

De permutatie-afleidingsregel is $\vdash C \Rightarrow \vdash C\pi$.

Definitie 103 *Permutatie-generalisatie Laat C_1 en C_2 een clausal form met n literals of $\blacksquare$ zijn. $C_1 \geq_p C_2$, C_1 is een permutatie-generalisatie van C_2, desda:*

- $C_1 = \square$;
- $C_2 = \blacksquare$;

- *er een substitutie θ en een permutatie π van $\{1, \ldots, n\}$ zijn zdd $C_2 = C_1\pi\theta$.*

Een substitutie-generalisatie is een bijzondere permutatie-generalisatie, namelijk de permutatie-generalisatie met permutatie $\{1, 2, 3, \ldots, n\}$.

Stelling 20 *Volgorde substitutie en permutatie Laat C een clausal form, θ een substitutie en π een permutatie zijn. Er geldt:*

(5.4) $C\pi\theta = C\theta\pi$

Bewijs 11 *Laat $C = L_1 \vee \ldots \vee L_n$.*

$$\begin{aligned} C\pi\theta &= (L_{\pi(1)} \vee \ldots \vee L_{\pi(n)})\theta \\ &= L_{\pi(1)}\theta \vee \ldots \vee L_{\pi(n)}\theta \\ &= C\theta\pi \end{aligned}$$

.

5.3.1 Strikte generalisatie

Permutatie-generalisatie is sterker dan substitutie-generalisatie. Als voor twee clausal forms C_1 en C_2 geldt dat $C_1 \geq_s C_2$, geldt ook $C_1 \geq_p C_2$.

Stelling 21 *Ekwivalentie Laat C_1 en C_2 twee clausal forms zijn. Als $C_1 \sim_s C_2$ dan $C_1 \sim_p C_2$.*

Bewijs 12 *$C_1 \sim_s C_2$ is gedefinieerd als $C_1 \geq_s C_2$ en $C_2 \geq_s C_1$. Hieruit volgt $C_1 \geq_p C_2$ en $C_2 \geq_p C_1$, dus $C_1 \sim_p C_2$.*

In deze paragraaf wordt bewezen dat een strikte substitutie-generalisatie ook een strikte permutatie-generalisatie is. Dit betekent dat voor alle clausal forms C_1 en C_2 zdd $C_1 >_s C_2$ ook geldt $C_1 >_p C_2$.

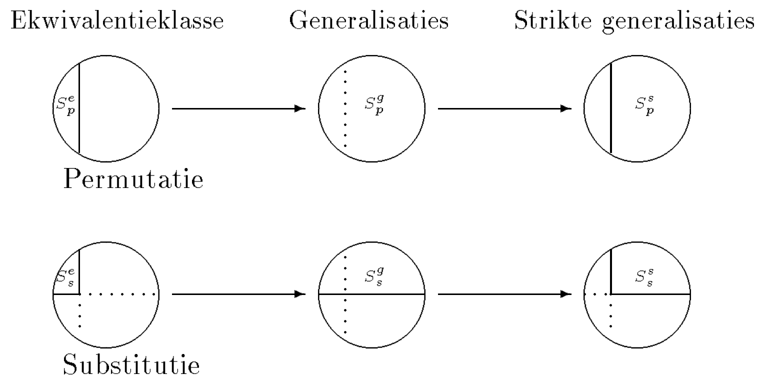
Figuur 5.3 geeft de strikte generalisaties van een clausal form weer:

- S_p^e, S_s^e: de permutatie- en substitutie-ekwivalenteklasse;
- S_p^g, S_s^g: de verzameling van alle permutatie- en substitutie-generalisaties;

  De permutatie-generalisaties worden gerepresenteerd door de cirkel. De ekwivalentieklasse is een deelverzameling van de generalisaties, het is immers een kwasi-ordening.

- S_p^s, S_s^s: de strikte permutatie- en substitutie-generalisaties;

  Strikte generalisaties zijn generalisaties die niet in de ekwivalentieklasse zitten.



Figuur 5.3: Strikte generalisatie

De substitutie-ekwivalentieklasse is een deelverzameling van de permutatie-ekwivalentieklasse, $S_s^e \subseteq S_p^e$. Bovendien is de verzameling substitutie-generalisaties een deelverzameling van de verzameling permutatie-generalisaties, $S_s^g \subseteq S_p^g$.

Als een strikte substitutie-generalisatie ook een strikte permutatie-generalisatie is, moet gelden dat $S_s^s \subseteq S_p^s$.

Dit is ekwivalent met de eis dat permutatie-ekwivalente clausal forms geen strikte substitutie-generalisaties zijn, ofwel dat de doorsnede van S_p^e en S_s^s leeg is.

Het is ook ekwivalent met de eis dat alle permutatie-ekwivalente clausal forms die een substitutie-generalisatie zijn, ook substitutie-ekwivalent zijn, ofwel dat S_s^e precies de doorsnede van S_s^g en S_p^e is.

Stelling 22 *Strikte permutatie-generalisatie Een strikte substitutie-ge-neralisatie is een strikte permutatie-generalisatie.*

Bewijs 13 *Evenals bij substitutie-generalisatie geldt bij permutatie-generalisatie:*

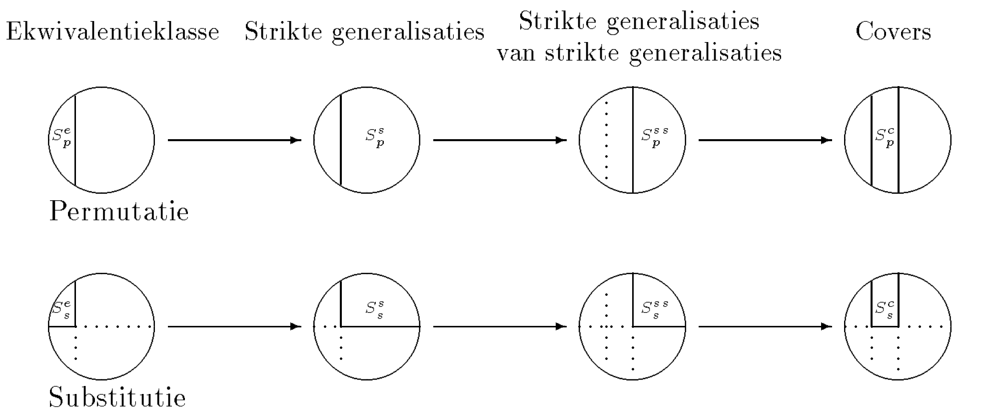
- *Als* $C_1 \sim_p C_2$ *dan is* $size(C_1) = size(C_2)$.
- *Als* $C_1 >_p C_2$ *dan is* $size(C_1) < size(C_2)$.

De size van alle clausal forms in S_s^s *is groter dan de size van een clausal form in* S_p^e, *dus is de doorsnede van* S_p^e *en* S_s^s *leeg.*

Hieruit volgt dat de twee clausal forms C_1 en C_2 permutatie-ekwivalent zijn desda C_1 en C_2 alleen verschillen in een permutatie van literals en een alfabetische variatie.

5.3.2 Cover

Figuur 5.4 geeft de permutatie- en substitutie-covers van een clausal form weer. De cirkel representeert wederom de verzameling van alle permutatie-generalisaties.



Figuur 5.4: Covers

- S_p^e, S_s^e: de permutatie- en substitutie-ekwivalenteklasse van een clausal form;
- S_p^s, S_s^s: de strikte permutatie- en substitutie-generalisaties;

  De generalisaties die niet in de ekwivalentieklasse zitten.
- S_p^{ss}, S_s^{ss}: strikte generalisaties van een strikte generalisatie;
- S_p^c, S_s^c: de permutatie- en substitutie-covers;

  De covers zijn het verschil van de strikte generalisaties en de strikte generalisaties van de strikte generalisaties.

In de vorige paragraaf is reeds bewezen dat $S_s^e \subseteq S_p^e$ en $S_s^s \subseteq S_p^s$. Hieruit volgt dat een strikte substitutie-generalisatie van een strikte substitutie-generalisatie ook een strikte permutatie-generalisatie van een strikte permutatie-generalisatie is, ofwel dat $S_s^{ss} \subseteq S_p^{ss}$.

In deze paragraaf wordt bewezen dat een substitutiecover ook een permutatiecover is, ofwel dat $S_s^c \subseteq S_p^c$.

Dit is ekwivalent met de eis dat een substitutie-cover geen strikte permutatie-generalisatie van een strikte permutatie-generalisatie is, ofwel dat de doorsnede van S_s^c en S_p^{ss} leeg is.

Dit is eveneens ekwivalent met de eis dat alle strikte permutatie-generalisaties van strikte permutatie-generalisaties die een strikte substitutie-generalisatie zijn, ook een strikte substitutie-generalisatie van een strikte substitutie-generalisatie zijn, ofwel dat de subset S_s^{ss} precies de doorsnede van S_s^s en S_p^{ss} is.

Alle strikte permutatie-generalisaties van strikte permutatie-generalisaties hebben een *size* die minstens twee groter is dan de *size* van de originele clausal form in C_p^e.

Voor alle clausal forms in S_e^c van type 1, type 2 en type 3 met $n = 1$ (het aantal vervangen funkties) geldt dat zij niet in S_p^{ss} kunnen voorkomen daar hun size kleiner is dan de size van alle clausal forms in S_p^{ss}. Voor substitutie-covers van type 3 die meerdere funktievoorkomens vervangen zal echter een alternatief bewijs moeten worden opgesteld.

Hier wordt een intuïtief bewijs gegeven, voor een formeler bewijs wordt verwezen naar [44].

Het bewijs bestaat uit twee stappen. Allereerst wordt aangetoond dat de substituties van een substitutie-cover in zekere zin alfabetische variaties zijn, het is niet mogelijk zo'n substitutie te splitsen in twee niet-triviale substituties zoals bijvoorbeeld de substitutie $\{x/f(f(z))\}$ kan worden gesplitst in $\{x/f(y)\}$ en $\{y/f(z)\}$ (omdat permutaties en substituties onafhankelijk zijn van elkaar). Uit stelling 20 volgt dat het niet mogelijk is dat een substitutie-cover geen permutatie-cover is.

Stelling 23 *Ekwivalentie Laat C_1 en C_2 twee clausal forms zijn zdd $C_1 \to_s C_2$ en θ_1 de substitutie zdd $C_1\theta_1 = C_2$. Voor een substitutie θ_2 en permutatie π zdd $C_1\theta_1 = C_1\theta_2\pi$ geldt dat de afbeelding θ_2 in zekere zin een alfabetische variatie is van θ_1.*

Hiermee wordt bedoeld dat als θ_1 een variabele vervangt door een term, er een variabele mee korrespondeert in θ_2 die dezelfde term (en evenveel termvoorkomens) vervangt.

Bewijs 14 *Dit volgt triviaal uit het feit dat de θ_1 alleen een substitutie van één variabele door een konstante, een reeds voorkomende variabele of een funktie met nieuwe verschillende variabelen kan zijn.*

Stelling 24 *Permutatie-cover Een substitutie-cover is ook een permutatie-cover.*

Bewijs 15 *Laat C_1 en C_2 twee clausal forms zijn. Stel dat C_2 een substitutie-cover ($C_1\theta = C_2$) en een strikte permutatie-generalisatie van een strikte permutatie-generalisatie ($C_1\theta_1\pi_1\theta_2\pi_2 = C_2$) van C_1 is.*

Er geldt $C_1\theta = C_1\theta_1\pi_1\theta_2\pi_2$ ofwel $C_1\theta = C_1\theta_1\theta_2\pi_1\pi_2$. Dan zijn er substituties (ongelijk herbenoeming variabelen) zdd $\theta = \theta_1'\theta_2'$. Dit is onmogelijk, uit deze tegenspraak volgt dat de premisse niet waar kan zijn: een substitutie-cover kan geen strikte permutatie-generalisatie van een strikte permutatie-generalisatie zijn.

Het blijkt dus dat substitutie-generalisatie en permutatie-generalisatie dicht bij elkaar liggen. De permutatie-covers kunnen worden berekend door gebruik te maken van de substitutie-covers; de voordelen van termstrukturen en termpartities bij het bepalen van generalisaties gelden ook voor de permutatie-kwasi-ordening.

5.3.3 Supremum

Het substitutie-supremum (en infimum) van twee clausal forms bestaat altijd en dus induceert de substitutie-kwasi-ordening een tralie. De permutatie-ordening blijkt echter geen tralie te induceren daar het permutatie-supremum niet altijd bestaat [44].

Stelling 25 *Permutatie-supremum Het permutatie-supremum van twee clausal forms bestaat niet altijd.*

Bewijs 16 *Laat C_1, C_2, C_3 en C_4 de volgende vier clausal forms zijn:*

- $C_1 = L_{1,1} \vee L_{1,2}$ *met* $L_{1,1} = P(f(a), f(b))$ *en* $L_{1,2} = P(g(c), g(c))$.
- $C_2 = L_{2,1} \vee L_{2,2}$ *met* $L_{2,1} = P(f(d), f(d))$ *en* $L_{2,2} = P(g(e), g(k))$.
- $C_3 = L_{1,2} \vee L_{1,1}$, *permutatie-ekwivalent aan* C_1.
- $C_4 = L_{2,2} \vee L_{2,1}$, *permutatie-ekwivalent aan* C_2.

C_3 en C_4 zijn, onder alfabetische variatie, de enige permutatie-ekwivalente clausal forms van C_1 en C_2 (naast C_1 en C_2 zelf).

Substitutie-suprema van twee aan C_1 en C_2 permutatie-ekwivalente clausal forms zijn bovengrenzen van het permutatie-supremum van C_1 en C_2.

- $C_1 \sqcup_s C_2 = P(f(x), f(y)) \vee P(g(z), g(u))$.
- $C_1 \sqcup_s C_4 = P(v, v) \vee P(x, y)$.

Het infimum van $C_1 \sqcup_s C_2$ en $C_1 \sqcup_s C_4$ is gelijk aan $\blacksquare$ omdat geen enkele permutatie van hen te unificeren is. $\blacksquare$ is geen bovengrens van C_1 en C_2, dus bestaat het permutatie-supremum niet.

De substitutie-suprema hoeven dus geen permutatie-suprema te zijn; Nienhuys-Cheng [44] noemt ze pseudo-suprema voor de permutatie-kwasi-ordening. De verzameling strikte bovengrenzen van een clausal form is eindig dus heeft deze verzameling wel minimale elementen, dit zijn de pseudo-suprema.

Merk op dat deze eigenschap het voordeel van termstrukturen en termpartities bij het bepalen van suprema doet verdwijnen.

5.4 Set

Set-generalisatie kan worden opgevat als de substitutie-variant voor clauses. Voor clausal forms C_1 en C_2 met korresponderende clauses C_1' en C_2' geldt dat C_1 een set-generalisatie van C_2 is desda er een substitutie θ is zdd $C_1'\theta = C_2'$.

Definitie 104 *Set-generalisatie Laat C_1 en C_2 een clausal form of $\blacksquare$ zijn. $C_1 \geq_e C_2$, C_1 is een set-generalisatie van C_2, desda:*

- $C_1 = \square$;
- $C_2 = \blacksquare$;
- *er een substitutie θ is zdd er voor alle literals L_1 in C_1 een literal L_2 in C_2 en er voor alle literals L_2 in C_2 er een literal L_1 in C_1 is zdd $L_1\theta = L_2$.*

De clause van een clausal form is de verzameling literals die voorkomen in de set-gereduceerde clausal form; de clause en deze clausal form bestaan uit evenveel literals.

Definitie 105 *Set-reduktie Laat C een clausal form zijn. De set-geredu-ceerde clausal form van C wordt verkregen door alle literal-voorkomens die meerdere malen voorkomen in de clausal form op één na te verwijderen.*

Het onderscheid tussen clausal forms met substitutie-generalisatie en clauses met set-generalisatie is gemaakt omdat de eerste mooiere (tralie-) eigenschappen heeft. In de volgende paragrafen zal worden aangetoond dat het bepalen van covers en suprema in de set-kwasi-ordening problematischer is.

5.4.1 Inverse substitutie

Bij clausal forms kan een inverse substitutie worden gerepresenteerd met behulp van een termpartitie. Iets dergelijks zou ook bij clauses (ofwel set-generalisatie) moeten worden opgesteld.

Het verschil met clausal forms is dat meerdere literals bij een substitutie op dezelfde literal kunnen worden afgebeeld. Zo is $\{P(x), P(y)\}$ een generalisatie van $\{P(a)\}$.

Een set-generalisatie is een substitutie-generalisatie van een clausal form die set-ekwivalent is. Een mogelijke oplossing is eerst deze set-ekwivalente clausal form te konstrueren en hier vervolgens de substitutie-generalisatie van te nemen, die bijvoorbeeld met een termpartitie kan worden gerepresenteerd. In het bovenstaande voorbeeld is deze set-ekwivalente clausal form $P(a) \vee P(a)$. Deze aanpak wordt door Nienhuys-Cheng [44] gehanteerd.

Een nadeel is dat twee termpartities van dezelfde clause niet meer vergeleken kunnen worden, omdat zij kunnen verwijzen naar verschillende set-ekwivalente clausal forms. Een andere mogelijkheid is iedere literal in de clause een index mee te geven (zoals in clausal forms gebruikelijk) en voor ieder voorkomen in de generalisatie ook een volgnummer. De literal $P(a)$ heeft dan index 1 en de literals $P(x)$ en $P(y)$ volgnummers $(1, 1)$ en $(1, 2)$.

De definities van termpartities kunnen dan op triviale wijze worden uitgebreid voor clauses.

Het aantal (niet set-ekwivalente) inverse substituties is bij clauses oneindig; daarom wordt meestal een begrenzing op de kompleksiteit van de inverse substitutie gelegd, bijvoorbeeld op het aantal literals.

Een probleem bij het naïef genereren van inverse substituties van clauses, bijvoorbeeld door inverse substituties van clausal forms van de clause te genereren, is dat vele inverse substituties set-ekwivalent zijn. Een heuristiek zou dit kunnen verbeteren, bijvoorbeeld geen literal twee keer voor laten komen als beide niet worden gemanipuleerd.

5.4.2 Strikte generalisatie

In deze paragraaf wordt bewezen dat een strikte substitutie-generalisatie een strikte set-generalisatie is. Het gevolg is dat de ekwivalentieklasse onder set-generalisatie gelijk is aan de alfabetische varianten van een clause.

Er wordt een alternatieve kompleksiteitsmaat ingevoerd, gebaseerd op het aantal literals in de clause van de clausal form en de size van de literals in de clause. Deze kompleksiteitsmaat is gebaseerd op de volgende twee waarnemingen:

- Bij een generalisatie kan het aantal literals niet afnemen, echter wel toenemen (tenzij de generalisatie gelijk is aan $\Box$).
- De *size* van een literal in de clause neemt bij een generalisatie niet toe. Als het aantal literals niet verandert neemt de *size* van de clausal form bij een strikte generalisatie af.

Definitie 106 *$Size_{set}$ Laat C een clausal form of $\blacksquare$ zijn. $Size(C)$ is gelijk aan:*

- $size_{set}(\Box) = \infty$;
- $size_{set}(\blacksquare) = 0$;
- *lit(C) = aantal literals in de set-gereduceerde clausal form van C;*
- *siz(C) = size van de set-gereduceerde clausal form van C;*
- $size_{set}(C) = lit(C) + 2^{-(1+siz(C))}$.

Stelling 26 *Strikte set-generalisatie Een strikte substitutie-generalisatie is een strikte set-generalisatie.*

Bewijs 17 *Als bij een generalisatie het aantal literals toeneemt, dan neemt de $size_{set}$ toe want $lit(C)$ domineert $2^{-(1+siz(C))}$ (ligt altijd tussen nul en één). Als het aantal literals gelijk blijft dan neemt $siz(C)$ af en dus neemt $2^{-(1+siz(C))}$ toe. Dus geldt er:*

- *Als $C_1 \sim_e C_2$ dan is $size_{set}(C_1) = size_{set}(C_2)$.*
- *Als $C_1 >_e C_2$ dan is $size_{set}(C_1) > size_{set}(C_2)$.*

De $size_{set}$ van alle strikte substitutie-generalisaties is groter dan de $size_{set}$ van de set-ekwivalente clausal forms (vergelijk figuur 5.3).

5.4.3 Cover

In de substitutie-kwasi-ordening heeft $\Box$ in de duale ordening covers, namelijk clausal forms met als termen alleen verschillende variabelen. In de set-ordening geldt dit echter niet: er is altijd een clausal form tussen een clausal form en de lege clausal form.

Helaas blijkt dat een substitutie-cover geen set-cover is, er zijn substitutie-covers die strikte set-generalisaties van strikte set-generalisaties zijn (vergelijk figuur 5.4).

Stelling 27 *Set-cover Een substitutie-cover hoeft geen set-cover te zijn.*

Bewijs 18 *Beschouw het volgende tegenvoorbeeld, C_1, C_2 en C_3 zijn clausal forms.*

- $C_1 = P(a) \vee P(f(b)) \vee P(a)$;
- $C_2 = P(a) \vee P(f(b)) \vee P(x)$;
- $C_3 = P(a) \vee P(f(b)) \vee P(f(y))$;

C_2 is een substitutie-cover van C_1, maar C_3 is een strikte set-generalisatie van C_1 en C_2 is een strikte set-generalisatie van C_3.

Uit het voorbeeld blijkt dat dit ook geldt voor Horn-clausal forms.

De ordeningen op clauses zijn fundamenteel anders dan de ordeningen op clausal forms. Shapiro [56] dacht een ordening op clausal forms te kunnen generalisaren naar clauses maar van der Laag [23] toonde aan dat deze generalisatie niet korrekt is.

5.4.4 Supremum

Het feit dat permutatie-supremum niet hoeft te bestaan, betekent nog niet dat het set-supremum ook niet hoeft te bestaan. Helaas is ook dit het geval.

Stelling 28 *Set-supremum Het set-supremum hoeft niet te bestaan.*

Bewijs 19 *Beschouw de clausal forms van stelling 25. Wederom is het set-infimum van $C_1 \sqcup_s C_2$ en $C_1 \sqcup_s C_4$ gelijk aan $\blacksquare$, dus is er geen set-supremum.*

5.4.5 Ketens

In tegenstelling tot de permutatie-kwasi-ordening bestaan er wel oneindige stijgende ketens bij de set-kwasi-ordening. Een voorbeeld is de volgende oneindige stijgende keten, gebonden door $P(x, x)$ [47]:

- $C_1 = \{P(x_0, x_1)\}$
- $C_i = C_{i-1} \cup \{P(x_{i-1}, x_i)\}$

Ook zijn er oneindig dalende ketens; het voorbeeld van de oneindig dalende keten onder substitutie-generalisatie is ook een oneindige keten onder set-generalisatie.

5.5 Subsumptie

Subsumptie-generalisatie is een uitbreiding van set-generalisatie; ook het verwijderen van literals is toegestaan.

Definitie 107 *Subsumptie Laat C_1 en C_2 twee clausal forms zijn. $C_1 \geq_u C_2$, C_1 is een subsumptie-generalisatie van C_2, desda één van de volgende twee waar is:*

- $C_2 = \blacksquare$;
- *er een substitutie* θ *is zdd er voor alle literals* L_1 *in* C_1 *een literal* L_2 *in* C_2 *is zdd* $L_1\theta = L_2$.

Stelling 29 *Subsumptie Laat* C_1 *en* C_2 *twee clauses zijn.* $C_1 \geq_u C_2$, C_1 *subsumes*[8] C_2, *desda één van de volgende twee waar is:*

- $C_2 = \blacksquare$;
- *er een substitutie* θ *is zdd* $C_1\theta \subseteq C_2$.

Bewijs 20 *Volgt direkt uit de definitie.*

5.5.1 Strikte generalisatie

De ekwivalentieklasse onder subsumptie-generalisatie is gelijk aan clauses die kunnen worden gereduceerd tot dezelfde clause (onder alfabetische variatie).

Definitie 108 *Gereduceerd Laat* C *een clause zijn.* C *is gereduceerd desda voor alle* C' *waarvoor geldt* $C' \subseteq C$ *en* $C \sim_u C'$, *geldt dat* $C = C'$.

Als voor twee clauses C_1 en C_2 geldt dat $C_1 \sim_u C_2$ dan hoeven C_1 en C_2 geen alfabetische varianten te zijn, bijvoorbeeld $C_1 = \{P(a)\}$ en $C_2 = \{P(a), P(x)\}$.

Plotkin [47] heeft aangetoond dat voor twee gereduceerde clauses C_1 en C_2 met $C_1 \sim_u C_2$ wel geldt dat C_1 en C_2 alfabetische varianten zijn. Iedere ekwivalentieklasse heeft dus een uniek gereduceerd lid (onder alfabetische variatie).

De volgende algoritme geeft een gereduceerde subset C_2 van C_1 zdd $C_2 \sim C_1$ door redundante literals te verwijderen [47]:

1. $C_2 := C_1$;
2. Vind een L in C_2 en een substitutie θ zdd $C_2\theta \subseteq C_2 \backslash \{L\}$. Stop als er niet zo'n L meer is;
3. $C_2 := C_2\theta$. Ga naar 2.

In de vorige paragraaf is bewezen dat een strikte substitutie-generalisatie ook een strikte set-generalisatie is. Het blijkt echter dat hij geen strikte subsumptie-generalisatie hoeft te zijn.

Stelling 30 *Strikte subsumptie-generalisatie Een strikte substitutie-ge-neralisatie hoeft geen strikte subsumptie-generalisatie te zijn.*

Bewijs 21 *Laat* C_1 *en* C_2 *de volgende clausal forms zijn:*

- $C_1 = P(a) \vee P(x)$;

[8] In de latere literatuur vaak θ-subsumes genoemd.

- $C_2 = P(a) \vee P(a)$.

C_1 is een strikte substitutie-generalisatie (en een strikte set-generalisatie) maar geen strikte subsumptie-generalisatie van C_2 want beide zijn te reduceren tot $P(a)$.

Het vinden van strikte subsumptie-generalisaties zou kunnen worden gesplitst in drie delen: eerst alle permutatie-ekwivalente clauses genereren (onder een bepaalde restriktie; het zijn er immers oneindig veel), vervolgens set-generalisaties vinden van deze permutatie-ekwivalente clauses (wederom onder een restriktie) en vervolgens deze clauses reduceren om te testen of zij ekwivalent zijn.

Deze methode is zeer reken-intensief; het reduceren van clauses kost veel tijd en vele clauses kunnen op meerdere manieren worden gekonstrueerd. Het probleem dat bij set-generalisatie in de vorige paragraaf al optrad, steekt hier in nog desastreuzere vormen zijn kop op [23].

5.5.2 Supremum en infimum

Het supremum en infimum onder de subsumptie-kwasi-ordening bestaan altijd, deze kwasi-ordening induceert dus een tralie. Het infimum is gelijk aan de disjunktie van twee clausal forms, net als het infimum van twee wff onder logische implikatie.

Definitie 109 *Selektie Laat $S = \{C_i \mid i = 1, \ldots, n\}$ een verzameling van clauses en $K = \{L_i \mid i = 1, \ldots, n\}$ een verzameling van literals zijn. K is een selektie van S desda $L_i \epsilon C_i$ $(1 \leq i \leq n)$ en alle literals in S sup-vergelijkbaar zijn.*

Plotkin [47] heeft aangetoond dat iedere verzameling clauses een supremum ongelijk aan de lege formule heeft desda die verzameling een selektie heeft.

De natuurlijke partiële ordening (op clauses) is een tralie:

- Ekwivalentieklasse: $[C] = \{C_i \mid C_i \sim C\}$;
- Ordening: $[C_1] \geq_u [C_2]$ desda $C_1 \geq_u C_2$;
- Supremum: $[C_1] \sqcup [C_2] = [C_1 \sqcup C_2]$;
- Infimum: $[C_1] \sqcap [C_2] = [C_1\theta \cup C_2]$;

  θ is een herbenoeming van variabelen die de variabelen van C_1 en C_2 doet verschillen.

De tralie is niet-modulair en er zijn oneindige stijgende en dalende ketens.

5.6 Logische implikatie

Definitie 110 *Logische generalisatie Laat C_1 en C_2 twee clauses zijn. $C_1 \geq_l C_2$, C_1 is een logische generalisatie van C_2, desda $C_1 \models C_2$.*

Subsumptie is zwakker dan logische implikatie. Beschouw het volgende voorbeeld van Niblett [35].

- $C_1 = \{\neg P(f(x)), P(x)\}$
- $C_2 = \{\neg P(f(f(x))), P(x)\}$.
- $C_1 \models C_2$;
- $C_1 \not\geq_u C_2$.

Van twee clauses C_1 en C_2 kan in eindige tijd worden bepaald of C_1 een subsumptie-generalisatie (of een zwakkere generalisatie) van C_2 is; subsumptie is beslisbaar [35]. Logische generalisatie voor clauses is echter slechts semi-beslisbaar, alleen als een clause een logische generalisatie is kan dit in eindige tijd worden bepaald. Een uitzondering hierop is logische generalisatie voor Horn-clauses.

Stelling 31 *Beslisbaarheid logische generalisatie Voor Horn-clauses is logische generalisatie beslisbaar.*

Bewijs 22 *Dit is een gevolg van de kompleetheid van SLD-resolutie. Het aantal te doorzoeken SLD-bomen is eindig. Zie Nibblett [35].*

5.6.1 Supremum

Het supremum onder subsumptie-generalisatie bestaat; dit blijkt echter niet het geval te zijn voor logische generalisatie.

Niblett [35] geeft een voorbeeld van twee Horn-clauses H_1 en H_2 die geen supremum hebben:

- $H_1 = \{P(x), \neg P(f(f(x)))\}$;
- $H_2 = \{P(x), \neg P(f(f(f(x))))$;
- $G_1 = \{P(x), \neg P(f(x))\}$;
- $G_2 = \{P(x), \neg P(f(f(y)))$.

G_1 en G_2 zijn twee bovengrenzen van H_1 en H_2. Er is geen specifiekere bovengrens dan G_2 (want de positieve literal moet $P(x)$ bevatten) maar $G_1 \not\models G_2$.

Het lijkt intuïtief voor de hand liggend dat dit niet voor clauses geldt, maar daar heb ik geen bewijs voor kunnen vinden.

In een deelverzameling van de clause logika, namelijk zonder funktiesymbolen, is logische generalisatie wel beslisbaar en bestaat het supremum [35]. Het uitdrukkingsvermogen is dan echter sterk afgenomen.

Hoofdstuk 6

Generalisatie met achtergrondtheorie

6.1 Inleiding

In lerende systemen zijn generalisaties van clauses zonder achtergrondtheorie niet erg realistisch. In dit hoofdstuk worden de konsekwenties van achtergrondkennis besproken.

Plotkin [48] hanteert in zijn model subsumptie als generalisatie. Dit model is uitgebreid voor een achtergrondtheorie bestaande uit grondliterals en later voor een willekeurige achtergrondtheorie in zijn promotie-these.

Buntine [9] laat voor Horn-clauses zien dat de subsumptie zonder achtergrondtheorie en Plotkins relatieve subsumptie meerdere nadelen hebben en definieerde een tussenweg, gegeneraliseerde subsumptie.

Om relatieve en gegeneraliseerde subsumptie te kunnen implementeren moet het resolutiemechanisme worden geïnverteerd. Muggleton en Buntine [31] definieerden hiertoe een aantal operatoren.

Matwin en Lapointe [29] breidden deze inverse resolutie uit, namelijk gebaseerd op logische generalisatie in plaats van subsumptie, zodat ook rekursieve clauses kunnen worden afgeleid.

6.2 Ekwivalentie

Maher [28] heeft onderzocht wanneer twee Horn-clause programma's ekwivalent zijn. Hierbij wordt onderscheid gemaakt tussen subsumptie-ekwivalentie en logische ekwivalentie. Dit geeft enig inzicht in het onderscheid tussen subsumptie en logische ekwivalentie.

Om te testen of programma's ekwivalent zijn worden zij herschreven naar een logisch ekwivalent, gereduceerd programma, de kanonieke vorm (*canonical form*).

Definitie 111 *Kanonieke vorm Laat P een logisch programma zijn. De kanonieke vorm van P is het programma waarin alle clauses gereduceerd zijn (zie* *definitie 108) en waarin alle clauses die worden gesubsumeerd door een andere clause in P verwijderd zijn.*

De programma's P_1 en P_2 hebben dezelfde kanonieke vorm desda iedere clause in P_1 een subsumptiegeneralisatie van een clause in P_2 is en vice versa. Bovendien geldt dat P_1 and P_2 dezelfde kanonieke vorm hebben desda $T_{p_1} = T_{p_2}$ (zie definitie 65), ofwel als precies dezelfde clauses met resolutie kunnen worden afgeleid.

Een tautologie is niet erg interresant bij deduktie of induktie. Als de tautologieën ook worden verwijderd blijft de zwakke kanonieke vorm over.

Definitie 112 *Zwakke kanonieke vorm Laat P een logisch programma zijn. De zwakke kanonieke vorm van P is de kanonieke vorm waaruit alle tautologieën zijn verwijderd. Een tautologie is een clause waarin een complementair paar voorkomt.*

Twee logische programma's P_1 and P_2 hebben dezelfde zwakke kanonieke vorm desda iedere clause in P_1 logisch wordt geïmpliceerd door een clause in P_2 en vice versa.

6.3 Relatieve subsumptie

Het subsumptie generalisatiemodel wordt uitgebreid met achtergrondkennis; clauses zijn generalisaties relatief aan een logische theorie.

6.3.1 Grondtheorie

Plotkin [48] heeft een model van relatieve subsumptie-generalisatie opgesteld, waarbij de achtergrondtheorie bestaat uit een konjunktie van grondliterals (waarmee hij het axiomatisatie-probleem omzeilt).

Definitie 113 $\overline{T}$ *Laat T een verzameling grondliterals zijn. $\overline{T} = \{\neg L \mid L \epsilon T\}$.*

Met behulp van deze definitie kan relatieve subsumptie-generalisatie worden gedefinieerd.

Definitie 114 $\geq_u$ *relatief aan T Laat T een theorie en C_1 en C_2 twee clauses zijn. $C_1 \geq_u C_2(T)$, C_1 is een subsumptie-generalisatie van C_2 relatief aan T, desda $\{C_1\} \geq_u \{C_2\} \cup \overline{T}$.*

Een voorbeeld van subsumptie en relatieve subsumptie. Een mogelijke interpretatie is dat $Q(f(y))$ wordt verklaard door $P(x)$:

- $\{\neg P(x), Q(f(y)\} \geq_u \{\neg P(a), Q(f(y))\}$
- $\{\neg P(x), Q(f(y)\} \not\geq_u \{Q(f(y))\}$
- $\{\neg P(x), Q(f(y)\} \geq_u \{Q(f(y))\}(\{P(a)\})$

Definitie 115 *$\sim_u$ relatief aan T Laat T een theorie en C_1 en C_2 twee clauses zijn. $C_1 \sim_u C_2(T)$ desda $C_1 \geq_u C_2(T)$ en $C_2 \geq_u C_1(T)$.*

Net als bij generalisatie zonder achtergrondtheorie kan worden geanalyseerd wanneer twee literals of twee clauses ekwivalent zijn. Bovendien kan het algoritme om gereduceerde clauses te vinden worden aangepast. De resultaten komen sterk overeen met de uitkomsten in het vorige hoofdstuk.

Stelling 32 *Relatieve ekwivalentie literals Laat T een theorie en L_1 en L_2 twee literals zijn. $L_1 \sim_u L_2(T)$ desda één van de volgende twee waar is:*

- *L_1 en L_2 zijn alfabetische varianten van elkaar;*
- *L_1 en L_2 subsumen $\overline{T}$.*

Bewijs 23 *Zie [48].*

Definitie 116 *Gereduceerd Laat T een theorie en C een clause zijn. C is gereduceerd, relatief aan T, desda voor alle C' waarvoor geldt dat $C' \subseteq C$ en $C' \sim_u C(T)$, geldt dat $C = C'$.*

Stelling 33 *Relatieve ekwivalentie clausal forms Laat T een theorie zijn en C_1 en C_2 twee relatief aan T gereduceerde clausal forms. Als $C_1 \sim C_2(T)$ dan zijn C_1 en C_2 alfabetische varianten.*

Bewijs 24 *Zie [48].*

Laat T een theorie en C_1 een clause zijn. De volgende algoritme [48] geeft een gereduceerde clause C_2, relatief aan T, van C_1 zdd $C_2 \sim_u C_1(T)$.

1. $C_2 = C_1$;
2. Vind een L in C_1 en een substitutie θ zdd $C_2\theta \subseteq (C_2 \backslash \{L\}) \cup \overline{T}$. Als zo'n L niet bestaat, stop dan.
3. $C_2 = C_2\theta$. Ga naar 2.

Behalve clauses kunnen ook verzamelingen van clauses, logische theorieën, gereduceerd worden [48].

Definitie 117 *Gereduceerd Laat H_1 en T twee logische theorieën zijn. H_1 is gereduceerd, relatief aan T, desda voor alle logische theorieën H_2 waarvoor geldt $H_2 \subseteq H_1$ en $H_2 \sim_u H_1(T)$, geldt dat $H_2 = H_1$.*

De volgende algoritme van Plotkin [48] geeft een gereduceerde theorie H_2, relatief aan de theorie T, van de theorie H_1 zdd $H_2 \sim_u H_1(T)$.

1. $H_2 = H_1$;
2. Vind een clause C in H_2 zdd $H_2 \backslash \{C\} \geq_u \{C\}(T)$. Als zo'n C niet bestaat, stop dan.
3. $H_2 = H_2 \backslash \{C\}$. Ga naar 2.

6.3.2 Theorie

Vervolgens onderzoekt Plotkin subsumptie relatief aan niet-grondtheorieën [9]. Deze ordening wordt door hem gegeven zonder een algoritme om deze generalisaties ook werkelijk te berekenen.

Definitie 118 *Relatieve subsumptie Laat T een theorie en C_1 en C_2 twee clauses zijn. $C_1 \geq_u C_2(T)$ desda er een substitutie θ is zdd $T \vdash_{res} C_1\theta \rightarrow C_2$.*

Dit is ekwivalent met de eis dat er een substitutie θ is zdd $C_1\theta \subseteq C_3$ en $T \vdash_{res} C_2 \leftrightarrow C_3$. [35]

Dit is niet gelijk aan logische generalisatie daar de resolutie-afleidingsregel niet kompleet is. De volgende definitie en stelling geven het verschil tussen subsumptie en logische implikatie weer.

Definitie 119 *C-afleiding Laat C een clause zijn. Een C-afleiding is een afleiding waarbij resolutie de enige afleidingsregel is en waarbij C maximaal één keer voorkomt in de resolutieboom.*

Stelling 34 *Relatieve subsumptie Laat T een theorie en C_1 en C_2 twee clauses zijn. $C_1 \geq_u C_2(T)$ desda C_2 een tautologie is of er een C_1-afleiding van een clause C_3 is en $C_3 \geq_u C_2$.*

Bewijs 25 *Zie [48] en [35].*

6.3.3 Gegeneraliseerde subsumptie

Buntine [9] definieert model-theoretisch gegeneraliseerde subsumptie voor Horn-clauses. Dit is eveneens een uitbreiding van subsumptie zonder achtergrondtheorie en wederom ontbreekt een algoritme om deze generalisaties te berekenen [35].

Het verschil met relatieve subsumptie is dat de positieve literal hetzelfde moet zijn. Een Horn-clause kan worden gezien als een beschrijving van een concept, waarbij de positieve literal het verklaarde concept weergeeft en de negatieve literals de verklaring. Een generalisatie beperkt zich dan tot hetzelfde concept.

Allereerst moet het begrip dekken gedefinieerd worden. Een clause dekt een atoom als deze clause direkt verantwoordelijk is voor het afleiden van dat atoom. Dit is het geval als er een substitutie is zdd de positieve literal van de clause gelijk wordt aan de literal en de negatieve literals van de clause waar zijn.

Definitie 120 *Dekken Laat $C = \{L_0, \ldots, L_n\}$ een Horn-clause zijn met L_0 een positieve en de overige L_i negatieve literals, L een positieve literal en I een interpretatie. C dekt L relatief aan I desda er een substitutie θ bestaat zdd $L_0\theta = L$ en $L_i\theta$ $(1 \leq i \leq n)$ waar zijn in I.*

Een clause is een subsumptie-generalisatie als het minstens dezelfde atomen dekt (vergelijk figuur 4.1).

Definitie 121 *Subsume Laat T een theorie en C_1 en C_2 twee Horn-clauses zijn. $C_1 \geq_b C_2(T)$ desda voor iedere interpretatie I waarin T waar is en alle positieve grondliterals L, C_1 L relatief aan I dekt als C_2 dat doet.*

Deze modeltheoretische definitie heeft de volgende ekwivalente syntaktische tegenhanger [35] voor de clauses C_1 en C_2. Er moet een substitutie θ_1 van de positieve literal van C_1 zijn zdd hij gelijk is aan de positieve literal van C_2. Bovendien moet elke grondsubstitutie van iedere negatieve literal van C_1 door de negatieve literals van C_2 afgeleid kunnen worden, ofwel (semantisch) er moet aan de voorwaarden van clause C_1 voldaan zijn als aan de voorwaarden van clause C_2 voldaan is[1].

Stelling 35 *Gegeneraliseerde subsumptie Laat $C_1 = \{L_{1,0}, \ldots, L_{1,n}\}$ en $C_2 = \{L_{2,0}, \ldots, L_{2,m}\}$ twee (apart gestandaardiseerde) Horn-clauses zijn met L_0 een positieve en de overige L_i negatieve literals en T een theorie. $C_1 \geq_b C_2(T)$ desda:*

- *er een minimale substitutie θ_1 bestaat zdd $L_{1,0}\theta_1 = L_{2,0}$.*
- *voor iedere grondsubstitutie θ_2 voor C_2 die nieuwe konstanten – die niet in C_1, C_2 of T voorkomen – substitueert, het waar is dat:*

$$(6.1) \quad \{T \cup \bigcup_{i \leq m} \{\neg L_{2,i}\theta_2\}\} \vdash_{sld} \bigwedge_{i \leq n} \neg L_{1,i}\theta_1\theta_2$$

Bewijs 26 *Zie [35].*

Gegeneraliseerde subsumptie is semi-beslisbaar want de bovenstaande bewijstheoretische variant is afhankelijk van SLD-resolutie[2].

De relatie tussen gegeneraliseerde subsumptie en logische implikatie is [9]:

Stelling 36 *Gegeneraliseerde subsumptie en logische implikatie Laat C een clause en T een programma zijn. Als C geen tautologie is geldt dat $T \models C$ desda er een clause C_1 in T is zdd $C_1 \geq_b C(T)$.*

Dit geldt dus alleen als C_1 een clause in T is, wat over het algemeen niet het geval is. Dit onderscheid lijkt door Rouveirol [55] over het hoofd gezien te zijn. Een afleidingsprocedure heet bij haar inversie-kompleet als het alle logische generalisaties af kan leiden met dezelfde positieve literal. De door haar beschreven afleidingsprocedure is mijn inziens echter helemaal niet kompleet: de beschreven procedure kan $P(x) \vee \neg P(f(x))$ niet afleiden van $P(x) \vee \neg P(f(f(x)))$. Het is slechts een implementatie van gegeneraliseerde subsumptie.

[1] Merk op dat het lijkt alsof L_1 en L_2 zijn omgedraaid. Dit wordt veroorzaakt door de negatie-tekens in de formule.

[2] In stelling 31 is bewezen dat logische implikatie voor Horn clauses (SLD-resolutie) beslisbaar is; dit geldt echter alleen als de theorie slechts één clause bevat!

6.3.4 Vergelijking relatieve en gegeneraliseerde subsumptie

In paragraaf 6.3.2 is aangetoond dat een clause C_1 een subsumptie-generalisatie van clause C_2 relatief aan T is desda er een C_1-afleiding van een clause C_3 is en $C_3 \geq_u C_2$, ofwel dat C_1 hoogstens één maal voorkomt in een ontkenning van $T \models C_1 \rightarrow C_2$.

Voor gegeneraliseerde subsumptie geldt dat C_1 een generalisatie van C_2 relatief aan T is desda C_1 als wortel voorkomt in de afleidingsboom van de ontkenning van $T \models C_1 \rightarrow C_2$. Gegeneraliseerde subsumptie is dus een speciaal geval van relatieve subsumptie.

Niblett [35] geeft het volgende voorbeeld dat gegeneraliseerde subsumptie zwakker is dan relatieve subsumptie voor de atomen L_1, L_2, L_3 en P:

- $(L_1 \rightarrow P) \geq_u (L_2 \rightarrow L_3)(P)$
- $(L_1 \rightarrow P) \not\geq_b (L_2 \rightarrow L_3)(P)$

Het model van Buntine is semantisch beter gemotiveerd [35]. Het volgende voorbeeld van Buntine [9] laat zien waarom gegeneraliseerde subsumptie beter werkt dan relatieve subsumptie[3].

- $C_1 = Kat(x) \rightarrow Klein(x)$;
- $C_2 = Pluizig(x) \wedge Kat(x) \rightarrow Knuffeldier(x)$;
- $C_3 = Kat(x) \rightarrow Huisdier(x)$;
- $C_4 = Klein(x) \wedge Pluizig(x) \wedge Huisdier(x) \rightarrow Knuffeldier(x)$.

Clause C_1 is algemener dan clause C_2 relatief aan C_3 en C_4 volgens Plotkin, maar niet volgens Buntine. C_1 is een beschrijving van het concept $Klein$, een onderscheid dat door Plotkin niet wordt gemaakt (en ook niet kan worden gemaakt omdat Plotkin zichzelf niet restrikteert tot Horn-clause logika).

6.3.5 Supremum

In het geval van achtergrondkennis is er niet altijd een supremum van twee clauses. Plotkin [48] bewijst dit voor het algemene geval en Niblett [35] voor Horn-clauses.

Niblett definieert de volgende Horn-clauses en bewijst dat er geen supremum van C_1 en C_2 relatief aan T bestaat:

- $C_1 = P_1(f(a))$
- $C_2 = P_1(g(a))$
- $T = \{P_2(f(x), y), P_2(g(x), y)\}$

[3] Predikaatnamen zijn voor de duidelijkheid als woorden in plaats van hoofdletters weergegeven.

- $G_i = P_2(x, g_1(x)) \wedge \ldots \wedge P_2(x, g_i(x)) \rightarrow P_1(x)$ voor $i \geq 0$

Alle clauses G_i zijn bovengrenzen van C_1 en C_2, een kleinste bovengrens is er niet omdat er altijd een kleinere G_i is (met grotere index i).

Plotkin bewijst dat het supremum, relatief aan een grondtheorie, wel altijd bestaat. Dit is een gevolg van het feit dat de Herbrand interpretaties dan eindig zijn [9]. Voor een theorie van grondatomen is dit supremum gelijk aan het supremum zonder achtergrondtheorie [48].

Als het supremum niet bestaat is niet bekend hoe ver moet worden gegeneraliseerd; het zou noodzakelijk kunnen zijn de logika uit te breiden met een top-down besturingsmechanisme als in `MIS` van Shapiro [35].

6.4 Inverteren van resolutie

Relatieve subsumptie en gegeneraliseerde subsumptie definiëren een (semi-beslisbare) ordening van clauses relatief aan achtergrondkennis, maar Plotkin [48] en Buntine [9] geven niet aan hoe – gegeven een clause – generalisaties gevonden kunnen worden. Daar de ordening gedefinieerd is met behulp van de resolutie-algoritme, ligt het voor de hand te trachten deze algoritme te inverteren.

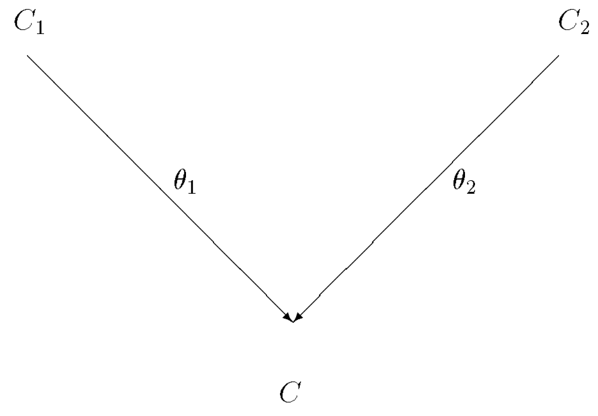
Met behulp van inverse resolutie kan wederom een partiële ordening worden vastgelegd. Merk op dat deze definitie algemener is dan gebruikt bij relatieve en gegeneraliseerde subsumptie; het is hier ook toegestaan dat C_1 meerdere keren in de resolutieboom voorkomt. Aan de andere kant ontbreekt hier de mogelijkheid literals te droppen (er geldt niet dat $P(a) \geq_{res} P(a) \vee Q(b)(\{\})$).

Definitie 122 $\geq_{res}$ *Laat T een theorie en C_1 en C_2 clauses zijn. $C_1 \geq_{res} C_2(T)$, C_1 is een resolutie-generalisatie van C_2 relatief aan T, desda C_2 afgeleid kan worden van T en C_1 met behulp van resolutie, ofwel $T \cup \{C_1\} \vdash_{res} C_2$.*

Voor Horn-clauses kan een identieke definitie worden gegeven voor $\geq_{sld}$, inverteren van sld-resolutie.

Gegeven de drie ordeningen – relatieve subsumptie, gegeneraliseerde subsumptie en komplete inverse resolutie – is nu de vraag hoe generalisaties kunnen worden gevonden. Resolutie wordt opgedeeld in een aantal resolutiestappen en het ligt voor de hand ook het inverteren van resolutie op te delen in een aantal inverse resolutiestappen. Deze aanpak blijkt wel bij relatieve subsumptie maar niet bij algemene inverse resolutie te werken.

Muggleton en Buntine [31] baseren hun inverse resolutie op relatieve subsumptie en definieren een aantal operatoren, de V- en W-operatoren. De V-operator inverteert één resolutiestap en de W-operator is in staat om zelf predikaten *uit te vinden* (in plaats van heuristisch te *ontdekken*).



Figuur 6.1: Resolutiestap

Een eenvoudige versie van het model is geïmplementeerd in het lerende systeem `CIGOL`. Ling [26] laat zien dat de meeste lerende systemen, gebaseerd op inverse resolutie, als de systemen van Banerji [1], Rouveirol en Puget [53], Ling [26] en Muggleton [32], gebaseerd zijn op varianten van V- en W-operatoren.

Matwin en Lapointe baseren zich op komplete inverse resolutie om rekursieve clauses te vinden. Hun aanpak is niet gebaseerd op inverse resolutiestappen maar op analyse van termen.

6.4.1 V-operatoren

Figuur 6.1 representeert een resolutiestap, waarbij C de resolvente van C_1 en C_2 is. De V-operator inverteert deze afleiding door uit C_1 en C een (of alle) C_2 te berekenen.

Als er wordt geresolveerd op literal L_1 in C_1 en L_2 in C_2 dan wordt de resolvente [31]:

(6.2) $C = (C_1 - \{L_1\})\theta_1 \cup (C_2 - \{L_2\})\theta_2$

Alvorens deze stap wordt uitgevoerd worden er soms eerst faktors van C_1 en C_2 genomen (zie definitie 49), waarbij het dus kan voorkomen dat er meerdere literals van C_1 of C_2 niet in de resolvente voorkomen. Het uitbreiden van een inverse resolutiestap met faktorisatie is overigens triviaal.

Seperability assumption

Muggleton en Buntine [31] introduceren de V- en W-operatoren onder de *seperability assumption*, wat wil zeggen dat de geresolveerde clauses, naast de literal waarop wordt geresolveerd, geen gezamelijke literals bevatten. In de volgende paragraaf worden de konsekwenties van het laten vallen van deze aanname geanalyseerd.

Onder de *seperability assumption* kan er niet op meerdere literals van C_1 en C_2 worden geresolveerd en speelt het faktor-probleem niet.

Definitie 123 *Seperability assumption Laat C_1 en C_2 twee clauses zijn met literal L_1 in C_1 en L_2 in C_2 en C de resolvente $(C_1 - \{L_1\})\theta_1 \cup (C_2 - \{L_2\})\theta_2$. De seperability assumption is geldig desda $(C_1 - L_1)\theta_1$ en $(C_2 - L_2)\theta_2$ geen gezamelijke literals bevatten.*

De V-operator konstrueert C_2 gegeven C en C_1. Er kan worden geresolveerd op een positief of een negatief literal in C_1. In het eerste geval wordt de V-operator *absorption* genoemd, in het tweede geval *identification*.

De clause C is redundant geworden; in feite vervangt de clause C_2 hem. Bovendien is het mogelijk dat ook andere clauses van de theorie nu redundant zijn geworden. Zo leveren de V-operatoren de mogelijkheid een theorie kompakter te maken [32].

Door formule 6.2 te herschrijven worden de C_2 gevonden (onder de *separability assumption*), waarbij de inverse substitutie het aantal literals kan doen toenemen:

(6.3) $C_2 = (C - (C_1 - \{L_1\})\theta_1)\theta_2^{-1} \cup \{L_2\}$

Er wordt geresolveerd op literals L_1 en L_2, dus geldt $\neg L_1\theta_1 = L_2\theta_2$, ofwel:

(6.4) $L_2 = \neg L_1\theta_1\theta_2^{-1}$

Dus geldt er (door 6.4 te substitueren in 6.3):

(6.5) $C_2 = ((C - (C_1 - \{L_1\})\theta_1) \cup \{\neg L_1\}\theta_1)\theta_2^{-1}$

In deze algemene formulering zijn er drie keuzen die gemaakt kunnen worden bij een resolutiestap: L_1, θ_1 en θ_2^{-1}.

Voor de meest specifieke V-operator geldt dat C_1 een literal is, ofwel $C_1 = \{L_1\}$, en θ_2^{-1} leeg is. Dan geldt:

(6.6) $C_2 = C \cup \{\neg L_1\}\theta_1$

Zonder seperability assumption

Muggleton [32] laat vervolgens de *seperability assumption* vallen.

Laat C_1 en C_2 twee clauses zijn die resolveren op literals L_1 en L_2 en $C_1' = (C_1 - \{L_1\})\theta_1$ en $C_2' = (C_2 - \{L_2\})\theta_2$. Er geldt dan $C = C_1' \cup C_2'$. Laat vervolgens $D = C_1' - C_2'$. Dan is:

(6.7) $C_2 = (C - D)\theta_2^{-1} \cup \{L_2\}$

En gegeven $L_2 = \neg L_1\theta_1\theta_2^{-1}$ blijkt:

(6.8) $C_2 = ((C - D) \cup \{\neg L_1\}\theta_1)\theta_2^{-1}$

Als deze vergelijking wordt vergeleken met de eerder gevonden vergelijking met *seperability assumption* (6.5) blijkt dat bij de konstruktie van C_2 uit C en C_1 nu ook nog de keuze bestaat van een D, onder de restriktie dat $\emptyset \subseteq D \subseteq C_1'$. Als $D = C_1'$ dan is de oplossing gelijk, maar als een D wordt gekozen waarin enige literals van C_1' niet voorkomen, dan komen zij wel in $(C - D)$ en dus in C_2 voor. Ofwel, er is dan een resolutiestap geïnverteerd waarbij naast L_1 nog meer overeenkomstige literals in C_1 en C_2 voorkomen.

De meest specifieke oplossing kan worden gevonden door de volgende twee aannamen: $D = \emptyset$ en $\theta_2^{-1} = \{\}$. De meest specifieke oplossing, genoteerd met $C_2\downarrow$, is:

(6.9) $C_2 \downarrow = (C \cup \{\neg L_1\}\theta_1)$

Merk op dat iedere C_2 een subsumptie-generalisatie van $C_2\downarrow$ is, $C_2 \geq_u C_2\downarrow$. Dit geldt tevens als eerst faktors van C_1 en C_2 worden genomen. Al is de hier beschreven inverse resolutiestap niet kompleet [40] in die zin dat C de resolvente van alle mogelijke faktors van C_1 en C_2 is, wel geldt dat $C_2\downarrow$ altijd de meest specifieke oplossing is.

Niet iedere subsumptie-generalisatie van $C_2\downarrow$ is ook een C_2 zdd C de resolvente van C_1 en C_2 is.

De substitutie θ_1 wordt grotendeels bepaald door het feit dat $(C_1 - L_1)\theta_1 \subseteq C$. Alleen de variabelen die wel in L_1 maar niet in $(C - L_1)$ voorkomen zijn onbepaald. Als zulke variabelen niet bestaan worden de clauses sterk generatief genoemd. Voor sterk generatieve clauses geldt dus dat de meest specifieke C_2 uniek bepaald is.

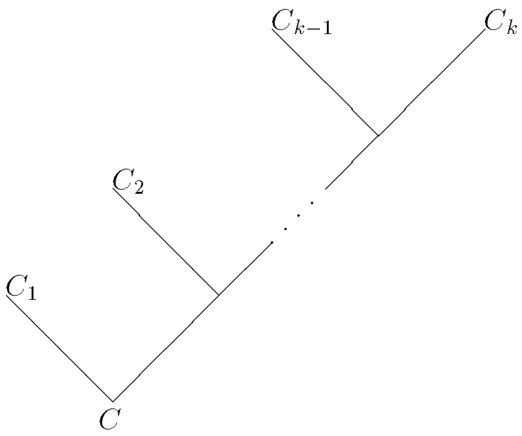
Definitie 124 *Sterk en zwak generatief Laat T een verzameling clauses zijn. T is sterk generatief desda iedere variabele in een clause in minstens twee literals van die clause voorkomt.*

Een logisch programma T is zwak generatief als de variabelen in de positieve literal van een Horn-clause ook in een negatieve literal van die clause voorkomen.

De restriktie van sterk generatief lijkt te sterk, alleen de literal waarop wordt geresolveerd mag geen variabelen bevatten die in de rest van de clause niet voorkomen.

Zwak generatieve Horn-clauses zou de meest specifieke oplossing van de *absorbtion* V-operator (waarbij immers op een positieve literal in C_1 wordt geresolveerd) uniek bepalen. Ook zwak generatief is nog een vrij sterke restriktie: de bekende Horn-clause definitie van append is bijvoorbeeld niet zwak generatief (een lijst wordt opgevat als een geneste rij funkties $lijst(kop, staart)$):

- $Append([], x, x)$;
- $Append([v \mid x], [y], [v \mid z]) \vee \neg Append(x, y, z)$.



Figuur 6.2: Lineaire afleiding

6.4.2 De funktie V^n

Een lineaire afleiding (zie paragraaf 60) is de opeenvolging van een aantal resolutiestappen, waarbij de resolvent van de vorige resolutiestap wordt geresolveerd met een clause. Met behulp van de V-operator kan een lineaire afleiding worden geïnverteerd.

Definitie 125 *V^n De funktie V^n beschrijft de verzameling van meest specifieke clauses die gekonstrueerd kunnen worden door een iteratieve toepassing van de V-operatoren. De funktie $V^n(P,C) : Programma's \times Clauses \rightarrow Programma's$ is door Muggleton [32] als volgt gedefinieerd:*

$$\begin{array}{rrl} & V^0(P,C) = & \{C\} \\ (6.10) & V^n(P,C) = & V^{n-1}(P,C) \cup \{(C' \cup \{\neg L\}\theta) \mid \\ & & L \epsilon C_{1,2} \epsilon P \wedge C' \epsilon V^{n-1}(P,C) \wedge (C_{1,2} - \{L\})\theta \subseteq C'\} \end{array}$$

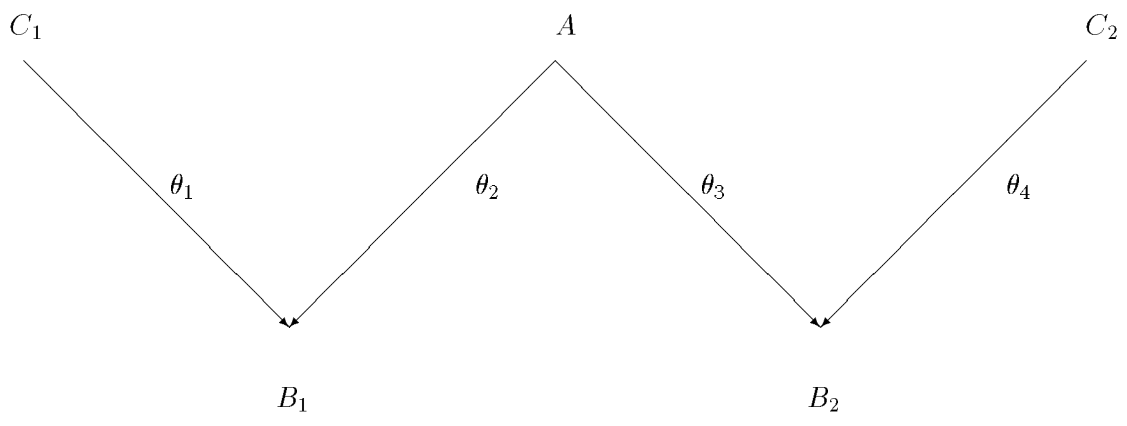
De afsluiting $V^*(P,C)$ is gelijk aan $V^1(P,C) \cup V^2(P,C) \cup \ldots$.

Stelling 37 *Kompleet De funktie V^n is kompleet met betrekking tot lineaire afleiding van sterk generatieve logische programma's.*

Laat T een sterk generatieve logische theorie zijn en $\{C_1,\ldots,C_{k-1}\}$ een deelverzameling van T. De lineaire afleiding resolveert (zie figuur 6.2):

- *in de eerste stap C_{k-1} met een onbekende clause C_k;*
- *vervolgens in iedere resolutiestap de resolvente van de vorige stap met een C_i.*

Voor iedere lineaire afleiding met diepte k geldt dat $V^k(P,C)$ een clause $C_k\downarrow$ bevat zdd C_k een subsumptie-generalisatie van $C_k\downarrow$ is. Er geldt dan dat C_k een subsumptie-generalisatie van $V^k(P,C)$ is.



Figuur 6.3: W-operatoren

Bewijs 27 *Zie [32].*

Als twee clauses C_1 en C_2 lineaire afleidingen van dezelfde onbekende clause C_k zijn, dan is C_k een subsumptie-generalisatie van $V^k(P, C_1)$ en $V^k(P, C_2)$. Bovendien geldt dat het subsumptie-supremum relatief aan T van de clauses C_1 en C_2 gelijk is aan het het subsumptie-supremum van $(\cup V^*(P, C_1))$ en $(\cup V^*(P, C_2))$ [32].

Deze clauses kunnen oneindig zijn; zelfs als zij logisch gereduceerd zijn. Plotkin tracht deze clause zelf toe te voegen aan de theorie, Muggleton en Buntine zoeken de meest kompakte konsistente subsumptie-generalisatie.

6.4.3 W-operatoren

Door twee resolutiestappen tegelijkertijd uit te voeren ontstaat de W-operator die het mogelijk maakt nieuwe predikaten te introduceren, zie figuur 6.3 [31].

Hierbij wordt aangenomen dat C_1 en C_2 resolveren op dezelfde literal L in A. De W-operator konstrueert C_1, A en C_2 gegeven B_1 en B_2. Als de literal L negatief is wordt de W-operator *intra-construction* genoemd, anders *inter-construction*.

Daar er wordt geresolveerd op de literal L komt deze niet meer voor in B_1 en B_2. Ofwel, als een W-operator wordt toegepast kan er een nieuw predikaat worden geïntroduceerd.

De W-operator kan worden uitgebreid naar meer dan twee simultane resolutiestappen. Laat BB een verzameling van B_i en CC een verzameling van C_i zijn, L een negatieve literal in A zdd B_i de resolvente van A en C_i is, resolverend op literal L. Dan is B_i gelijk aan:

(6.11) $B_i = (A - \{L\})\theta_{A,i} \cup (C_i - \{L_i\})\theta_{C,i}$

Als de C_i literals zijn, dus $C_i = \{L_i\}$, dan geldt:

(6.12) $B_i = (A - \{L\})\theta_{A,i}$

Ofwel:

(6.13) $A = B_i\theta^{-1}_{A,i} \cup \{L\}$

Waarbij $B_i\theta^{-1}_{A,i}$ het set-supremum van de B_i is. Als de C_i geen literals zijn kan gekozen worden voor het subsumptie-supremum van de B_i.

De W-operator is verder te generaliseren door aan te nemen dat de B_i lineaire afleidingen van A zijn. A is dan het relatieve subsumptie-supremum van de B_i.

Nadat A is gekonstrueerd kunnen de C_i worden gekonstrueerd door de V-operator toe te passen.

6.4.4 Lapointe en Matwin

Bij inverse resolutie, gebaseerd op subsumptie, kunnen clauses worden gevonden die maximaal één keer in de resolutieboom voorkomen (zie stelling 34). Dit betekent dat rekursieve clauses alleen kunnen worden gevonden als er twee voorbeelden met slechts één rekursie-stap verschil worden aangeboden.

Het blijkt dat Prolog-programma's meestal grotendeels uit rekursieve clauses bestaan. De voorbeelden die worden gegeven door Shapiro [56] en Muggleton en Buntine [31] van – door hun systemen `MIS` en `CIGOL` – geleerde clauses zijn ook rekursief. Een operator die rekursieve clauses kan vinden is een nuttige uitbreiding op het raamwerk van Muggleton en Buntine.

Inverse resolutie en subsumptie

Matwin en Lapointe [29] geven de volgende Horn-clauses als voorbeeld van de tekortkomingen van subsumptie:

- $C_1 = Append([], x, x)$;
- $C_2 = Append([a, b, c], [1, 2], [a, b, c, 1, 2])$;
- $C_3 = Append([v \mid x], [y], [v \mid z]) \vee \neg Append(x, y, z)$.

C_1 en C_3 zijn Horn-clauses die de funktie *Append* beschrijven; een lijst wordt opgevat als een geneste rij funkties $lijst(kop \mid staart)$.

Gegeven C_1 en C_2 is het vrij eenvoudig te zien dat C_3 waarschijnlijk de gezochte clause is. Inverse resolutie gebaseerd op subsumptie kan dit niet vinden; inverse resolutie gebaseerd op logische implikatie wel. De eerste kan met C_2 niets aanvangen; hij is niet bruikbaar.

Inverse resolutie en logische implikatie

Rekursieve clauses kunnen worden gevonden door termen in plaats van hele clauses te analyseren; Matwin en Lapointe noemen dit *subunifikatie*, een afkorting van *subterm unifikatie*.

Bij rekursie wordt een variabele elke rekursiestap vervangen, zo ontstaan geneste funkties. Een voorbeeld hiervan is de rij die de natuurlijke getallen representeert in de axiomata van Peano: $0, s(0), s(s(0)), s(s(s(0))), \ldots$.

Een operator die rekursieve clauses kan vinden moet deze geneste funkties herkennen. Matwin en Lapointe noemen termen die een aantal maal een variabele vervangen *genererende termen*. Hun operator is erop gebaseerd deze genererende termen in voorbeelden te zoeken en daarna clauses te konstrueren die zoveel mogelijk resolutiestappen nodig hebben om de voorbeelden af te leiden.

Hoofdstuk 7

Induktief leren

7.1 Inleiding

In de vorige drie hoofdstukken zijn algemene principes en wiskundige eigenschappen van generalisatie-ordeningen op formulae binnen de clause logika behandeld. In dit hoofdstuk wordt gekeken hoe generalisatie-ordeningen kunnen worden toegepast binnen induktief lerende systemen.

Generalisatie is een onderdeel, misschien een klein onderdeel, van een lerend systeem. In de praktijk zijn vragen over de zoekruimte, de mogelijke hypothesen, de besturingsmechanismen en de interaktie met een leraar van groter belang. Het is hierbij interessant om in het achterhoofd te houden dat uit het werk van Gold blijkt dat logische theorieën in principe ook geleerd kunnen worden zonder gebruik te maken van een generalisatiemechanisme [35, p. 132].

Bij het vinden van de oplossing van het model-afleidingsprobleem is het noodzakelijk een heuristiek te hanteren, daar het aantal mogelijke modellen (of logische theorieën) meestal onbegrensd is. De in het vorige hoofdstuk behandelde inverse resolutie en subsumptie zijn hier voorbeelden van.

Bij heuristieken draait het om een rechtvaardiging van hypothesen. Er kan onderscheid worden gemaakt tussen heuristieken die gebruik maken van de aard van de kennis in het systeem en algemeen toepasbare heuristieken, waarbij de laatste uiteraard de voorkeur hebben. Een praktische heuristiek van de laatste groep is te streven naar een zo kompakt mogelijke theorie.

Een heuristiek maakt meestal een afweging tussen het aantal gebruikte voorbeelden en benodigde rekentijd (efficiëntie) om een theorie te vinden.

Een voorbeeld van een heuristiek zijn de diagnostische of abduktieve taken [12], waarbij een hypothese een relatie tussen oorzaken en gevolgen legt. Het model van Plotkin valt hieronder.

Meestal wordt bij induktief leren een aantal voorbeelden door een algemene regel verklaard en wordt er dus geen onderscheid gemaakt tussen oorzaak en gevolg. Deze verklaring wordt binnen de logika gezien als logische konsekwentie [12] of als de zwakke variant hiervan, *subsumptie* [48].

Deze voorbeelden worden meestal één voor één door het lerende systeem geanalyseerd; induktieve systemen moeten incrementeel zijn. Als een nieuw voorbeeld wordt toegevoegd dat niet kan worden verklaard, dan wordt de theorie enigszins (maar niet te veel) aangepast.

Allereerst worden de algemene eigenschappen van induktief leren behandeld en wordt gekeken naar de voorstellen in de literatuur om de clause logika te beperken (tot bijvoorbeeld de Horn-clause logika) of juist uit te breiden (met bijvoorbeeld een orakel). Vervolgens wordt ingegaan op de gehanteerde beperkingen binnen de clause logika.

7.2 Algemeen

In deze paragraaf worden de voorbeelden, de talen, orakels en redeneermethoden besproken.

7.2.1 Voorbeelden

Een induktief lerend systeem krijgt als invoer een aantal voorbeelden. Deze voorbeelden kunnen positief of negatief zijn. Positieve voorbeelden moeten door het systeem kunnen worden afgeleid, negatieve voorbeelden mogen niet door het systeem worden afgeleid maar moeten wel konsistent zijn.

Een voorbeeld van een negatief voorbeeld is: *als de straten nat zijn, dan heeft het geregend.* Het is mogelijk dat de straten nat zijn en dat dit is veroorzaakt door regen, er kunnen echter ook andere oorzaken voor bestaan.

Flach [12] verbindt aan deze twee verschillende soorten voorbeelden ook twee verschillende soorten verklaringen. Het verklaren van positieve voorbeelden heet een sterke verklaring (voorbeelden moeten afleidbaar zijn) en van de negatieve voorbeelden zwakke verklaring (voorbeelden moeten konsistent zijn).

Een populair leermodel binnen de machine learning is *PAC-learning*: probably approximately correct learning. Uitgaande van statistische eigenschappen moet een lerend systeem meestal ongeveer goede antwoorden geven.

Binnen het model van de PAC-learning is het ook toegestaan foute voorbeelden in het systeem in te voeren. Dit is niet het geval binnen de meeste modellen binnen de logika, daar een lerend systeem gebaseerd op (monotone) logika hier slecht mee overweg kan.

PAC-learning is ontstaan binnen het connectionisme. Het ligt in de aard van het connectionisme wel goed mee overweg te kunnen met ineksakte kennis maar juist niet met eksakte kennis (zie paragraaf 2.3.2).

7.2.2 Taal

Het representatieformalisme van een lerend systeem kan worden onderverdeeld in de talen van de hypothesen, observaties en achtergrondkennis.

Hypothesen

In deze skriptie staat eerste-orde predikatenlogika (in de vorm van clause logika) centraal. Het gebruik van de clause logika als representatieformalisme binnen een (induktief) lerend systeem heeft echter enige nadelen.

De eerste-orde logika wordt (binnen deduktieve systemen[1]) als te algemeen gezien, omdat het resolutiemechanisme niet efficiënt te implementeren is. Plotkin [48] stelde zijn model op aan de hand van de clause logika maar heeft zijn model nooit geïmplementeerd. Wel gaf hij twee mogelijk beperkingen van de clause logika die zijn model beslisbaar maken.

De populairste beperking is de Horn-clause logika, waarbinnen gebruik kan worden gemaakt van definiete informatie en dat efficiënt geïmplementeerd kan worden met behulp van Prolog. Voorbeelden van systemen zijn `MIS` van Shapiro [56], `CIGOL` van Muggleton en Buntine [31], `CAP` van Hume en Sammut [19], `IRES` van Rouveirol en Puget [55] en `Logical Traces` van Ling [26]. De laatste twee systemen maken niet werkelijk gebruik van de zoekruimte van de Horn-clause logika, want voor wordt gezocht naar hypothesen worden de clauses geskolemiseerd (funkties worden vervangen door nieuwe predikaten, Rouveirol noemt dit *flattening*).

Ook de Horn-clause logika is voor velen nog te algemeen. Banerji [1] maakt bijvoorbeeld slechts gebruik van monadische funktiesysmbolen (funkties met één parameter).

Binnen kennissystemen wordt hoofdzakelijk gebruik gemaakt van propositielogika, vaak uitgebreid met een statistisch besturingsmechanisme (proposities krijgen een waarschijnlijkheidsfaktor). Ook `Duce`, de voorloper van `CIGOL`, maakt gebruik van de propositielogika [31].

Een eerste-orde logika heeft echter ook zijn beperkingen en er zijn verschillende uitbreidingen binnen de logika voorgesteld (naast uitbreidingen buiten de logika, bijvoorbeeld een orakel, die in de volgende paragraaf worden besproken).

Niet-monotone logika's zijn hoofdzakelijk toegepast binnen deduktieve logika. Induktief lerende systemen maken alleen gebruik van de CWA binnen de Horn-clause logika.

Andere populaire uitbreidingen binnen de deduktieve logika zijn modale operatoren en metalogische restrikties (bijvoorbeeld een tweede-orde logika) [13]. Modale operatoren kunnen bijvoorbeeld frame-eigenschappen toevoegen aan de logika [60]. Lerende systemen die hier gebruik van maken zijn niet geïmplementeerd.

Niblett [36] stelt een metalogische restriktie voor waarbij clauses een waarschijnlijkheidsfaktor krijgen. Deze waarschijnlijkheidsfaktor is niet gebaseerd op Bayes-statistiek (als gebruikelijk binnen kennissystemen en onderhevig aan veel kritiek) maar gebaseerd op een waarschijnlijkheidstralie. Het opstellen van zo'n tralie is echter niet eenvoudig.

[1] De met behulp van induktie verkregen systemen worden vervolgens deduktief gebruikt; het is dus van belang dat een representatieformalisme van induktieve systemen ook met deduktie goed overweg kan.

Observaties

De observatietaal is de verzameling van alle mogelijke voorbeelden. Zij wordt vaak beperkt tot grondclauses of zelfs grondliterals, bijvoorbeeld binnen `CIGOL` [31].

Shapiro [56] definieert een kombinatie van observatietaal en hypothesetaal als toegestaan (*admissable*) als alle observaties de hypothese uniek vastleggen. Dit betekent dat als de hypothese niet korrekt is, er altijd een observatie in de observatietaal is die de hypothese falsificeert. Voorbeelden van toegestane kombinaties zijn:

- Grondatomen (observaties) en Horn-clauses (hypothesen);
- Grondliterals (observaties) en clauses (hypothesen).

Dit geeft een andere, eksakte kijk op het verschil tussen clause logika en Horn-clause logika.

Achtergrondkennis

De achtergrondkennistaal – de verzameling van alle mogelijke clauses in een achtergrondtheorie – wordt in het model van Plotkin [48] beperkt tot feiten. Andere modellen trachten deze taal uit te breiden omdat een achtergrondtheorie in de praktijk uit clauses bestaat [9].

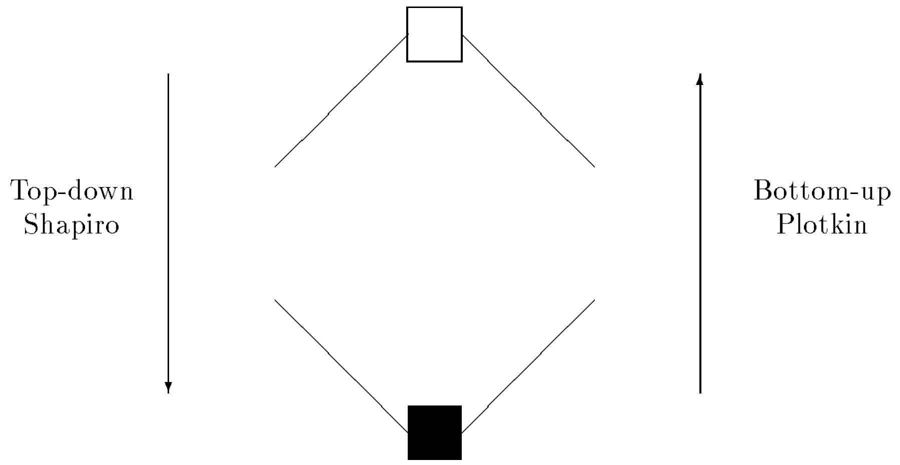
7.2.3 Orakel

De belangrijkste en meest gebruikte niet-logische uitbreiding op een lerend systeem is een orakel. Een orakel kan hypothesen testen op hun waarheidswaarde, in de praktijk meestal door middel van een menselijke leraar. Het toevoegen van een orakel verhoogt de efficiëntie van het lerende systeem wat betreft het benodigde aantal voorbeelden, maar de interaktie met een leraar is kostbaar, werkt fouten in de hand en is theoretisch niet wenselijk.

In `MIS` van Shapiro wordt een orakel gebruikt om te bepalen welke clause van het programma verantwoordelijk is als de *conjecture* te sterk is (zie paragraaf 2.3.4).

In `CIGOL` wordt een orakel gebruikt om de testen of een gegenereerde hypothese geen overgeneralisatie is. Als te algemene clauses binnen het systeem voorkomen zouden die weer gespecificeerd moeten worden en daarin voorziet `CIGOL` niet.

In het bijzonder wordt bij overgeneralisatie getest op konsistentie van het systeem. Dit is een algemeen probleem van de eerste-orde predikatenlogika als representatieformalisme, want konsistentie is slechts semi-beslisbaar.



Figuur 7.1: Induktief redeneren

7.2.4 Redeneermethoden

Er zijn twee verschillende redeneermethoden bij induktief leren, namelijk bottom-up en top-down, zie figuur 7.1. Induktief leren kan worden opgevat als een zoekstrategie in een partiële ordening [30]. Door gebruik te maken van een generalisatietralie kan de zoekruimte efficiënt worden doorzocht [9].

Bij bottom-up redeneren wordt een hypothese steeds verder gegeneraliseerd, zoals bijvoorbeeld in het model van Plotkin. Er wordt intensief gebruik gemaakt van de positieve voorbeelden, de negatieve voorbeelden zijn een bescherming tegen overgeneralisatie.

Bij top-down redeneren wordt een hypothese stap voor stap gespecialiseerd, zoals in `MIS` van Shapiro. Hierin wordt juist intensief gebruik gemaakt van de negatieve voorbeelden en zijn de positieve voorbeelden een bescherming tegen overspecialisatie.

Het top-down redeneren is de laatste paar jaar minder populair dan de bottom-up variant. Er kan minder goed gebruik worden gemaakt van relaties tussen twee clauses (zie de volgende paragraaf) en er wordt minder efficiënt gebruik gemaakt van een orakel. De in de literatuur beschreven implementaties van pure top-down systemen zijn ook veel minder grootschalig.

Het ligt voor de hand om te trachten beide redeneermethoden te kombineren, om zodoende zowel van positieve als van negatieve voorbeelden optimaal gebruik te kunnen maken. De filosofie van Popper volgend zou kunnen worden gesteld dat er geen kleine stapjes (b.v. covers) in de graaf moeten worden genomen, maar dat er vrijmoedig verzonnen theorieën moeten worden gegenereerd, die zowel te algemeen als te specifiek kunnen zijn.

Muggleton [33] stelt voor hypothesen te genereren die zo kompakt mogelijk zijn, met een minimale deskriptielengte (MDL). De motivatie van deze aanpak is dat theorieën zo eenvoudig mogelijk moeten zijn (zoals Popper al voorstelde). Dit maakt `CIGOL` veel efficiënter dan bijvoorbeeld het model van Plotkin en legt een link met de informatietheorie. De gegenereerde hypothesen worden in het orakel getest op overgeneralisatie dus `CIGOL` hanteert nog een puur bottom-up mechanisme.

Ling [26] hanteert Shapiro's *backtracing* algoritme indien het systeem een negatief voorbeeld af kan leiden. In zijn systeem wordt dus niet iedere hypothese op overgeneralisatie getest (dan zou dit immers niet nodig zijn).

Ishizaka [20] probeert gebruik te maken van bottom-up elementen in een top-down systeem door als positieve literal van gegenereerde Horn-clauses het supremum van feiten uit de achtergrondkennis te gebruiken. Dit supremum is echter meestal een overgeneralisatie en moet dus weer gespecialiseerd kunnen worden.

Wirth [63] kombineert bottom-up en top-down mechanismen in zijn `Learning by Failure to Prove (LFP)`. Het systeem bestaat uit twee verschillende stappen: de eerste stap (XGEN) bouwt top-down een gedeeltelijke bewijsboom (net als `MIS` van Shapiro) en de tweede stap hanteert inverse resolutie om de bewijsboom te kompleteren. De uitvoer van XGEN wordt niet (als in `MIS`) gebruikt als basis om via specialisaties de gezochte clause te vinden maar als invoer voor inverse resolutie.

Mitchell [30] probeert de bottom-up en top-down benadering te kombineren door zowel van algemeen naar specifiek als van specifiek naar algemeen te werken. Het verschil tussen beide, waarin zich de gezochte hypothese moet bevinden, noemt hij de *version space*.

7.3 Induktie in clause logika

De generalisaties van een logische theorie definieren een zoekruimte, deze zoekruimte is echter gigantisch groot. De beperking tot clauses reduceert deze zoekruimte aanzienlijk maar zij is nog altijd te groot.

De intuïtieve oplossing van dit probleem is te proberen een relatie te vinden tussen een aantal voorbeelden. Twee veelgebruikte relaties zijn het supremum en het inverteren van een resolutie-afleiding.

Restrikties op de zoekruimte van clauses restrikteren echter ook het aantal logische theorieën dat kan worden gegenereert.

7.3.1 Supremum

Binnen de generalisatietralie van alle wff binnen de clause logika is het supremum logische ekwivalent met de verzameling wff waarvan het supremum wordt genomen. Als echter een deeltralie hiervan wordt genomen, bijvoorbeeld de tralie van clauses, dan is het supremum meestal een logische generalisatie.

Een eenvoudig voorbeeld hiervan zijn de zwarte kraaien. Het wff-supremum bestaat simpelweg uit de vaststelling dat beide kraaien zwart zijn, het clause-supremum generaliseert dat alle kraaien zwart zijn.

- $C_1 = Zwart(kraai_1) \vee \neg Kraai(kraai_1)$
- $C_2 = Zwart(kraai_2) \vee \neg Kraai(kraai_2)$
- $C_1 \sqcup_{wff} C_2 = \begin{array}{l}(Zwart(kraai_1) \vee \neg Kraai(kraai_1)) \wedge \\ (Zwart(kraai_2) \vee \neg Kraai(kraai_2))\end{array}$
- $C_1 \sqcup_{clause} C_2 = Zwart(x) \vee \neg Kraai(x)$

Als er een clause is die alle (positieve) voorbeelden af kan leiden, dan is het supremum een instantie van die gezochte clause. Vaak zal er echter sprake zijn van overgeneralisatie, want er zijn vaak voorwaarden waaraan voldaan moet worden voor een feit kan worden afgeleid en een predikaat is meestal niet met één clause te beschrijven. Dit veroorzaakt de inefficiëntie van het model van Plotkin, dat slechts gebaseerd is op het vinden van suprema.

7.3.2 V-operatoren

Naast het supremum is een tweede relatie dat er een (mogelijk lineaire) afleiding bestaat van een clause. Een eenvoudig voorbeeld (waarbij één resolutiestap wordt geïnverteerd) is de afleiding van $p \rightarrow q$ uit p en q als in het volgende voorbeeld:

- Feit: *het regent.*
- Feit: *de straten zijn nat.*
- Induktie: *als het regent, dan zijn de straten nat.*

Zoals uit het voorbeeld blijkt kunnen met behulp van inverse resolutie kausale relaties worden gelegd, maar ook niet-kausale relaties zijn mogelijk.

Bij inverse resolutie gaat het niet alleen om het inverteren van resolutiestappen maar om de kombinatie met subsumbtie, er kunnen bijvoorbeeld ook literals worden verwijderd uit de clause (zie stelling 34).

Een inverse resolutiestap is een onderdeel van een inverse afleiding. De funktie V^n is hier een voorbeeld van, maar de beperking tot strikt generatieve clauses lijkt te beperkt. De inverse afleiding kan worden beperkt tot afleidingen waarin de gezochte clause als wortel voorkomt (gegeneraliseerde subsumptie), de clause op één willekeurige plaats voorkomt (relatieve subsumptie) of meerdere keren kan voorkomen (rekursieve clauses).

7.3.3 Een clause

Om het aantal generalisaties van een logische theorie te beperken wordt meestal één clause tegelijkertijd gegeneraliseerd. Dit levert een aanzienlijke beperking van de zoekruimte.

Dit is te verdedigen door het feit dat een generalisatie van een programma ook kan worden opgedeeld in een keten van generalisaties van clauses. Alle generalisaties van een logische theorie kunnen dus worden gevonden door elementaire stappen van generalisaties van clauses.

Zoals echter zojuist besproken worden clauses niet lukraak gegeneraliseerd, maar wordt er een relatie tussen een aantal voorbeelden gezocht. Sommige clauses kunnen dan niet meer gevonden worden. De W-operator is een voorbeeld van een operator die hier rekening mee houdt: meerdere clauses worden tegelijkertijd als hypothese gegenereerd.

Hoofdstuk 8

Samenvatting

In dit hoofdstuk worden de resultaten van het onderzoek samengevat en worden onderwerpen voor verder onderzoek aangegeven.

8.1 Samenvatting

Het onderzoek omvatte een analyse van induktieve modellen, ordeningen en toepassingen van ordeningen in induktieve modellen.

8.1.1 Induktieve modellen

De analyse van induktieve leermodellen wordt voorafgegaan door een beschrijving van filosofische kennis-, taal- en wetenschapstheorieën[1] en een beschrijving van het kader van induktief leren.

Een *kennistheorie* dient antwoord te geven op vier vragen: wat bestaat er, wat kan worden gekend, wat kan worden beweerd en hoe moet worden geleerd.

Een *taaltheorie* geeft antwoord op de vraag wat kan worden beweerd. Er zijn twee belangrijke stromingen binnen de taalfilosofie, het *atomisme* en het *holisme*. Het atomisme is gebaseerd op de *afbeeldingstheorie* en gaat ervan uit dat uitspraken (in de taal) en feiten (in de werkelijkheid) gereduceerd kunnen worden tot respektievelijk atomaire proposities en atomaire feiten. De afbeeldingstheorie stelt vervolgens dat atomaire proposities zodanig op atomaire feiten afgebeeld kunnen worden dat uitspraken kunnen worden begrepen door hun opbouw uit atomaire proposities en deze afbeelding op atomaire feiten. Het holisme stelt hier tegenover dat de reduktie tot atomen en de afbeelding op atomaire feiten niet mogelijk is en dat uitspraken alleen begrepen kunnen worden als onderdeel van een theorie (*taalspel* of *conceptueel schema*).

Een *wetenschapsfilosofische theorie* geeft antwoord op de vraag hoe moet worden geleerd. De westerse wetenschap is gebaseerd op het *empirisme*, de opvatting dat betrouwbare kennis gebaseerd moet zijn op ervaring. Drie belangrijke stromingen hebben de wetenschapsfilosofie deze eeuw bepaalt. De eerste, het *logisch*

[1] De filosofie gaat immers aan de andere wetenschappen vooraf.

positivisme, is gebaseerd op atomisme en induktieve logika en stelt dat kennis groeit door een accumulatie van feiten en dat induktie de enige juiste methode van leren is. De tweede stroming volgt *Poppers* kritiek op het logisch positivisme en stelt dat induktie niet van belang is voor wetenschappelijke theorieën. De groei van kennis is een gevolg van vrijmoedig verzonnen theorieën die aan kritische toetsen kunnen worden onderworpen. De laatste stroming, het *empirisch holisme* van Kuhn, is gebaseerd op holistische taaltheorieën stelde dat er voor toetsing en methodologische kriteria nauwelijks enige rol is weggelegd in de feitelijke ontwikkeling van wetenschap.

Het kader van induktieve leermodellen is het raakvlak van *kennisrepresentatie* en *machinaal leren*, beide onderdeel van *kunstmatige intelligentie*. Deze onderzoeksgebieden hebben een relatie met de hierboven genoemde filosofische theorieën.

Binnen de kunstmatige intelligentie zijn er twee stromingen. De eerste is *GOFAI*, *Good Old-Fashioned Artificial Intelligence*, gebaseerd op de afbeeldingstheorie en het atomisme. Het paradigma van intelligentie is het *oplossen van problemen* en de methode is gebaseerd op het *manipuleren van symbolen*. De tweede stroming is *connectionisme*, gebaseerd op het holisme. Het paradigma van intelligentie is *leren* en de methode is gebaseerd op *statistiek*.

Binnen de kennisrepresentatie zijn er drie stromingen, twee binnen GOFAI en één binnen connectionisme, ieder met een andere eenheid van kennis. De formalismen van de eerste zijn *logika's* en *produktiesystemen*, waarin de eenheid van kennis zo klein mogelijk wordt gekozen. De formalismen van de tweede stroming zijn *semantische netten* en *frames*, waarin de verbondenheid van kenniselementen kan worden aangegeven zodat grotere eenheden ontstaan. Het formalisme van de derde stroming is het *neurale netwerk*, waarin het hele systeem de kleinste eenheid van kennis is (de onderdelen van een neuraal netwerk representeren geen onderdeel van hetgeen het hele netwerk representeert).

Een *induktief leermodel* generaliseert voorbeelden tot hypothesen. Deze generalisatie kan worden onderverdeeld in een aantal stappen (is incrementeel). De *redeneermethode* bepaalt hoe de hypothesen worden gevonden, de twee basismethoden zijn *bottom-up* en *top-down*. Een bottom-up induktieve redenering bestaat uit een aantal generalisatiestappen, een top-down induktieve redenering begint met één generalisatiestap en bestaat verder uit een aantal specialisatiestappen.

Bekende bottom-up modellen zijn ontwikkeld door Plotkin en door Muggleton en Buntine en het bekendste top-down model is opgesteld door Shapiro. Veel moderne modellen trachten zowel top-down als bottom-up redeneringen te kombineren.

De induktieve leermodellen binnen een eerste-orde logika zijn alle beïnvloed door wetenschapsfilosofische theorieën, in het bijzonder het logisch positivisme (gebaseerd op induktie). Plotkin bespreekt ook een toepassing van elementen van de filosofie van Popper maar deze toepassing is pas succesvol bij Shapiro. Het model van Shapiro is gebaseerd op de interaktie van het opstellen van hypothesen en het toetsen van deze hypothesen. Bij het opstellen van hypothesen wordt nog wel gebruik gemaakt van induktie (wat volgens de radikalere Popper overbodig is in wetenschap).

8.1.2 Ordeningen

Een *ordening*, toegepast in induktieve modellen, is een rangschikking van formules van een logika die aangeeft hoe algemeen een formule is. Deze rangschikking kan alleen afhankelijk zijn van de formules zelf of gerelateerd zijn aan achtergrondkennis.

Een theorie in de wetenschap – een verzameling uitspraken – is een generalisatie als zij tenminste dezelfde uitspraken bevat (als een andere theorie). Een theorie in de logika – een verzameling formules – is een generalisatie als zij tenminste dezelfde formules kan afleiden.

Het *model-afleidingsprobleem* is het vinden van een model waarin een aantal formules waar zijn. Dit model kan worden gerepresenteerd met een logische theorie (een axiomatisatie), het model-afleidingsprobleem is dan ekwivalent aan het vinden van een generalisatie. Bij het opstellen van een *eindige* axiomatisatie van een model treedt het *axiomatisatieprobleem* op: twee verschillende axiomatisaties van hetzelfde model hoeven niet logisch ekwivalent te zijn.

Het begrip generalisatie kan worden vertaald naar het wiskundige begrip partiële ordening. In wiskundige termen bepaalt een generalisatiemodel hoe de zoekruimte van formules wordt bewandeld. Ordeningen op formules kunnen met behulp van wiskundige terminologie geanalyseerd worden. Centraal in de analyse staan de begrippen *bovengrens* (een algemenere formule), *cover* (de meest specifieke strikte bovengrens) en *supremum* (de meest specifieke bovengrens van een aantal formules).

De analyse van generalisatie-ordeningen van clauses zonder achtergrondtheorie omvat vijf verschillende ordeningen: *substitutie*, *permutatie*, *set*, *subsumptie* en *logische implikatie*.

Een substitutie-bovengrens is tevens een permutatie- en set-bovengrens, maar hoeft geen subsumptie-bovengrens te zijn. Een substitutie-cover is een permutatie-cover maar hoeft geen set-cover te zijn. Het substitutie-supremum en subsumptie-supremum bestaan, de suprema van de andere ordeningen hoeven niet te bestaan.

Clausal forms kunnen met een *termvoorkomenstruktuur* of met een *termstruktuur* worden gerepresenteerd en substitutie-bovengrenzen kunnen met een *inverse substitutie* of met een *termpartitie* worden gerepresenteerd. In een termstruktuur worden alle termvoorkomens apart gerepresenteerd en in een termvoorkomenstruktuur identieke termen bijelkaar. In een inverse substitutie wordt van ieder termvoorkomen aangegeven door welke variabele hij wordt vervangen en in een termpartitie zitten alle termen die door dezelfde variabele worden vervangen. In induktieve systemen worden clausal forms en substitutie-bovengrenzen als respektievelijk termstruktuur en inverse substitutie gerepresenteerd, maar door voor de andere mogelijkheden te kiezen kunnen de substitutie-bovengrenzen sneller worden berekend en vergeleken.

De generalisatie-ordeningen met achtergrondkennis maken gebruik van het (deduktieve) resolutiemechanisme. Een ordening kan op *subsumptie* en op *logische implikatie* gebaseerd zijn.

Als de ordening op subsumptie is gebaseerd mag de gezochte clause slechts één keer in de resolutie-bewijsboom voorkomen. Twee verschillende ordeningen zijn *relatieve subsumptie* van Plotkin, waarin de clause op een willekeurige plaats in de boom mag voorkomen, en *gegeneraliseerde subsumptie* van Buntine, waarin de clause alleen in de wortel mag voorkomen. Generalisaties kunnen in deze ordeningen worden gevonden door het inverteren van resolutie te splitsen in een aantal inverse resolutiestappen (de *V-operatoren*), zoals bij (deduktieve) resolutie ook gebruikelijk is.

De V-operatoren staan aan de basis van de *W-operatoren*, die het mogelijk maken nieuwe predikaten te ontdekken en zo de taal uit te breiden.

Als de ordening op logische implikatie is gebaseerd mag een clause meerdere keren voorkomen in de resolutie-bewijsboom en kan rekursie worden geïnverteerd. Generalisaties kunnen in deze ordening worden gevonden door termen te analyseren.

8.1.3 Toepassing van de ordeningen

Bij toepassing van generalisatie-ordeningen blijkt het nut van de wiskundige analyse. De induktieproblemen kunnen worden herschreven naar zoek-problemen in de generalisatie-hiërarchie. Hierbij komen verschillende andere problemen van induktieve modellen kijken.

De induktieve modellen zijn incrementeel, iedere stap wordt een *voorbeeld* aangeboden en wordt de theorie met behulp van dit voorbeeld enigszins aangepast. Een voorbeeld kan *positief* of *negatief* zijn. Een positief voorbeeld moet afleidbaar zijn en een negatief voorbeeld konsistent.

De *taal*, waarin de *hypothesen* worden uitgedrukt, is meestal niet een volledige eerste-orde logika maar een (deelverzameling van) Horn-clause logika. Naast de taal van de hypothesen zijn er ook de *talen* van de *voorbeelden* en de *achtergrondkennis* – verzameling van alle mogelijke voorbeelden respektievelijk achtergrondtheorieën – die vaak worden beperkt tot een deelverzameling van de taal van de hypothesen.

In induktieve modellen wordt vaak gebruik gemaakt van een *orakel*. In top-down modellen wordt een orakel gebruikt om een te algemene clause te vinden en in een bottom-up model om een hypothese op overgeneralisatie te testen.

Er zijn verscheidene pogingen gedaan de basis-redeneermethoden bottom-up en top-down te kombineren, zodat beter gebruik kan worden gemaakt van zowel positieve als negatieve voorbeelden. Ling [26] test (bottom-up verkregen) hypothesen niet op overgeneralisatie maar kan te algemene hypothesen later weer specificeren, Wirth [63] hanteert top-down informatie als invoer voor bottom-up generalisatie en Mitchell [30] werkt tegelijkertijd bottom-up en top-down.

De zoekruimte, gedefinieerd door een ordening, is gigantisch groot. Daarom worden voorbeelden vergeleken en wordt getracht een relatie tussen voorbeelden te vinden. Twee veel gebruikte relaties zijn het *supremum* van clauses [48] en een *inverse (lineaire) resolutie-afleiding* [31].

Alle besproken ordeningen zijn ordeningen op clauses in plaats van op theorieën, omdat het aantal mogelijke generalisaties van een theorie te groot is. De operatoren (behalve de W-operatoren) vinden slechts één clause, wat vaak een te sterke beperking is.

8.2 Verder onderzoek

In deze paragraaf worden enige interessante gebieden voor verder onderzoek aangegeven: de relatie van induktief leren met andere leermethoden (binnen machinaal leren), de verschillende toepassingen van inverse resolutie (in het bijzonder rekursieve clauses), nieuwe operatoren voor het uitbreiden van de taal en de kombinatie van bottom-up en top-down induktief redeneren.

8.2.1 Machinaal leren

Induktief logisch programmeren ligt op het raakvlak van logisch programmeren en machinaal leren [32]. De relatie met logisch programmeren is in deze skriptie vrij uitvoerig behandeld, de relatie met machinaal leren niet. Een analyse van deze relatie zou meer inzicht kunnen geven in de toepassing van statistiek en het gebruik van ordeningen binnen induktief leren.

Statistiek is de meest voor de hand liggende basis voor rechtvaardiging van geleerde hypothesen. In hoofdstuk 7 is kort ingegaan op de relatie van induktief leren met PAC-learning, gebaseerd op statistische eigenschappen. Binnen induktieve logika zijn ook enige pogingen gedaan statistische methoden toe te passen [48] [36], tot dusverre met weinig succes.

Ordeningen liggen aan de basis van induktief leren, zoals in het vorige hoofdstuk bleek. In andere leermodellen wordt ook vaak gebruik gemaakt van ordeningen. De relatie met deze modellen zou interessant kunnen zijn voor de toepassing van ordeningen binnen induktief lerende systemen.

8.2.2 Inverse resolutie

De wiskundige analyse van ordeningen van clauses zonder achtergrondtheorie is vrij uitputtend behandeld in hoofdstuk 5, de analyse van ordeningen met achtergrondtheorie in het erop volgende hoofdstuk is nog niet bevredigend. Bij de analyse werd uitgegaan van de V-operator (zoals gebruikelijk in de literatuur), maar zou naar mijn mening moeten uitgaan van de meer algemene definitie van inverse resolutie (zie definitie 122) en het laten vallen van literals.

De funktie V^n, die een aantal V-operatoren kombineert met subsumptie, is een speciaal geval van deze algemene definitie. Bovendien kan onderscheid worden gemaakt tussen de twee bouwstenen, namelijk inverse resolutie en het laten vallen van literals (of subsumptie).

De meeste systemen maken gebruik van verschillende restrikties op inverse resolutie (en eerste-orde logika). Het ligt voor de hand deze verschillende restrikties in detail te vergelijken.

Ling [26] en Rouveirol en Puget [53] hebben voorgesteld Horn-clauses voor generalisatie te skolemiseren. Deze representatie heeft dezelfde uitdrukkingskracht als de Horn-clause logika, maar de ordeningen op geskolemiseerde clauses verschillen van de ordeningen op clauses [3]. Een nadere analyse zou eksakt moeten aangeven wat deze verschillen zijn.

8.2.3 Rekursieve clauses

Traditionele systemen (gebaseerd op relatieve of gegeneraliseerde subsumptie) zijn gericht op het inverteren van lineaire afleidingen, waarin de gezochte clause één keer voorkomt. De afgeleide clauses leggen een relatie tussen verschillende concepten (bijvoorbeeld een kausale relatie).

In de praktijk werken een aantal clauses meestal gezamelijk, bijvoorbeeld het definieren van een concept in de Horn-clause logika. Een operator zou een aantal clauses tegelijkertijd moeten kunnen analyseren. Waarschijnlijk zal zo'n operator niet gebaseerd moeten zijn op de V-operatoren.

De beperking van V-operatoren blijkt uit het feit dat de huidige systemen goede voorbeelden nodig hebben om een onbekende theorie te vinden. Goede voorbeelden worden gekenmerkt door twee eigenschappen: de 'afstand' tussen twee clauses mag niet te groot zijn en de bodem van de rekursie moet bekend zijn.

Het vinden van generalisaties zal gebaseerd moeten zijn op een analyse van termen. Lapointe en Matwin [29] hebben hiervoor subunifikatie ontwikkeld, maar er is nog veel onderzoek mogelijk.

8.2.4 Uitbreiden van de taal

Het uitbreiden van de taal – het toevoegen van nieuwe predikaatsymbolen – definieert nieuwe concepten en is mogelijk dankzij de W-operatoren. De W-operatoren zijn echter niet erg krachtig, het aantal situaties waarin de taal wordt uitgebreid is niet erg groot [31]. Er zijn twee redenen om de taal te moeten kunnen uitbreiden.

Ten eerste maakt het introduceren van nieuwe predikaatsymbolen het (met een aantal operatoren) mogelijk clauses te vinden die anders niet gevonden zouden zijn. Dit geldt in het bijzonder binnen de Horn-clause logika, waarin iedere clause slechts één positieve literal bevat.

Ten tweede worden de logische theorieën kompakter, wat algoritmen efficiënter maakt.

Een voor de hand liggende vraag is of er andere operatoren verzonnen kunnen worden die ook in andere situaties nieuwe predikaatsymbolen kunnen introduceren.

Het onderzoek naar nieuwe operatoren zou geïnspireerd kunnen worden door het onderzoek op het gebied van abstraktie [14] (afbeeldingen op formules binnen een andere taal).

8.2.5 Kombinatie bottom-up en top-down

De basis-redeneermethoden zijn bottom-up en top-down, maar in moderne leermodellen wordt getracht deze twee methoden te kombineren, zodat van zowel positieve als negatieve voorbeelden optimaal gebruik kan worden gemaakt. De verschillende kombinatiemogelijkheden kunnen worden onderzocht en de gevolgen voor het gebruik van een orakel kunnen worden geanalyseerd.

Als bottom-up en top-down zoekmethoden worden gekombineerd, zijn er twee ekstreme mogelijkheden: de zoekruimte van boven en van beneden aflopen (van twee kanten af werken) of een hypothese genereren die zowel te algemeen als te specifiek kan zijn.

Mitchell [30] werkt van twee kanten in zijn `version spaces`, maar deze toepassing in een eerste-orde logika lijkt te uitgebreid.

Als een hypothese wordt verzonnen die zowel te algemeen als te specifiek mag zijn, is de vraag welke hypothese dan gekozen moet worden. Muggleton heeft voorgesteld de hypothese te kiezen die het kortst kan worden gerepresenteerd (MDL), maar wellicht zijn er ook andere heuristieken denkbaar.

Het model van Muggleton en Buntine is sterk afhankelijk van een orakel: de gebruiker moet aangeven of een afgeleide hypothese korrekt is. Het bestaan van oneindige ketens geeft aan dat het in een puur bottom-up model niet mogelijk is af te zien van een orakel, want iedere generalisatie kan te algemeen zijn.

Het kombineren van bottom-up en top-down methoden maakt het wel mogelijk minder afhankelijk te worden van dit orakel, want een (zoals later blijkt) te algemene hypothese kan gewoon aan de theorie worden toegevoegd en later, als hij te algemeen blijkt, weer worden gespecialiseerd.

Helaas zijn ook bottom-up modelllen afhankelijk van orakel, namelijk om vast te stellen welke clause te algemeen is als een theorie wordt gefalsificeerd. Het lijkt dus dat een orakel onmisbaar is.

Appendix A

Meta-logika

A.1 Inleiding

Bij de analyse van partiële ordeningen in hoofdstuk 5 is gebruik gemaakt van een formele analysemethode, voor het uiteindelijke resultaat is deze methode echter niet van belang. De methode is nog niet uitgekristalliseerd, daarom wordt hij in de appendix besproken.

Deze meta-logika is de opvolger (en deelverzameling) van een nog ambitieuzer opgezette meta-logika, de logika der logische strukturen. Deze logika zou een handvat moeten worden om ook de operatoren uit hoofdstuk 6 te analyseren, bijvoorbeeld de V- en W-operatoren.

In de volgende paragraaf wordt de meta-logika gedefinieerd en vervolgens wordt de toepassing voor de partiële ordeningen besproken. In de laatste paragraaf wordt de mogelijke uitbreiding naar de logika der logische strukturen besproken.

A.1.1 Doelstelling

Wat is het nut van de introduktie van deze meta-logika? Het is de laatste tien jaar mode om zoveel mogelijk nieuwe logika's te introduceren, en vaak is deze "abstraktie" slechts "abstrakte nonsens". Ik heb echter een aantal, naar mijn mening goede, redenen om dit toch te doen.

Ten eerste is de formele analyse de meest strikte bewijsvoering denkbaar. Er is geen "invoelend vermogen" nodig om de bewijsvoeringen te volgen. Een logika heeft een semantiek (de betekenis), die los staat van het formele bewijs.

Bovendien blijkt dat het probleem intuïtief duidelijker wordt door de clausal forms en clauses als verzamelingen te beschouwen. Daarom is er ook onderzocht of er gebruik kan worden gemaakt van de kategorieënleer, zie figuur 5.3 en 5.4.

Tot slot kunnen bepaalde eigenschappen van operatoren eenvoudiger worden verwoord en geklassificeerd. Een voorbeeld hiervan zijn de monotone operatoren, die in de volgende paragraaf zullen worden besproken.

Ieder gevonden bewijs kan ook worden gevonden en bewezen met behulp van standaard methoden; deze meta-logika is slechts een hulpmiddel bij het vinden van bewijzen. De meta-logika is in feite een trapleer die kan worden weggegooid als men omhoog is geklommen; de resultaten kunnen vrij eenvoudig worden terugvertaald naar de gebruikelijke bewijsvoeringen.

A.2 Partiële ordeningen

De in deze paragraaf besproken meta-logika definieert vijf verschillende verzamelingen: literal-lijsten, clausal forms, permutaties, clauses en gereduceerde clauses. De laatste vier representeren een ekwivalentieklasse onder een partiële ordening, zie hoofdstuk 5.

De logika bevat twee soorten modale operatoren: binaire operatoren die door middel van axiomata bovenstaande verzamelingen definiëren en unaire modale operatoren die eigenschappen als subsumptie, generalisatie etc. vastleggen.

Doel van deze meta-logika is een formeel bewijs te leveren voor de relaties van operatoren op de verschillende verzamelingen.

Er wordt bijvoorbeeld bewezen dat clausal forms clauses induceren bij strikte generalisaties en dat clausal forms permutaties induceren bij covers. Maar ook wordt bewezen dat clauses geen gereduceerde clauses induceren bij strikte generalisatie en dat permutaties geen clauses induceren bij covers.

Het nut van zo'n analyse is dat bij induktief redeneren eigenschappen van clauses benodigd zijn en slechts de eigenschappen van clausal forms bekend zijn. Als bij de V-operator de generalisaties van een clause worden berekend, kunnen hiervoor dan de generalisaties van clausal forms worden gebruikt of niet?

A.2.1 Meta-logika

De gebruikte meta-logika is een tweede-orde logika met, zoals gebruikelijk, een taal (alfabet en welgevormde formules), deduktietheorie (afleidingsregels en axiomata) en een semantiek. Het verschil met een eerste-orde logika is dat de konstanten en variabelen niet zomaar symbolen zijn, maar literals, clausal forms, substituties en variabelen hiervan.

Er wordt uitgegaan van de syntaks, dat wil zeggen dat overal syntaktische afleidingsregels voor worden opgesteld. Er kan dan worden bepaald of een bepaalde formule geldig is of niet.

Uiteraard is er ook een semantiek aan deze syntaks verbonden (het voordeel van logika immers) maar dit zal slechts als afgeleide worden gebruikt. De andere mogelijkheid was geweest de eigenschappen model-theoretisch te definiëren.

Daarom wordt ook logische implikatie (de laatste kwasi-ordening van hoofdstuk 5) niet gedefinieerd. Bij logische implikatie worden modellen van theorieën vergeleken, ofwel er wordt uitgegaan van de semantiek. Overigens kan logische implikatie ook syntaktisch worden gedefinieerd aan de hand van een komplete deduktietheorie. De kompleksiteit hiervan voert echter veel te ver, gezien de doelstelling van de meta-logika.

Syntaks

Definitie 126 *Alfabet Het alfabet van de meta-logika bestaat uit de volgende primitieven:*

- *Konstanten:*
  - *Literals, als gedefinieerd in de clause logika;*
  - *Substituties, idem;*
- *Variabelen:*
  - *Literals: L, eventueel met index;*
  - *Literal-lijsten: C, eventueel met index;*
  - *Substituties: θ, eventueel met index;*
- *Lijstsymbolen: $\langle$ en $\rangle$;*
- *Konnektieven, kwantoren, leestekens en unaire modale operatoren als gebruikelijk in de predikatenlogika[1];*
- *Een aantal identiteitsrelaties, $=$, $=^{cf}$, $=^{p}$, $=^{cl}$ en $=^{r}$. Dit zijn binaire modale operatoren die een relatie tussen twee literals of literal-lijsten weergeven.*

Een modale operator is een betekenisloze term, het zijn geen "truth-functional operators" als bijvoorbeeld $\wedge$. Voor een modale operator moeten definities (axiomata) worden opgesteld om te kunnen afleiden of een wff met een operator erin geldig is.

Definitie 127 *L-term Een literal-term (l-term) is:*

- *Een literal (konstante of variabele);*
- *Een l-term gevolgd door een substitutie (konstante of variabele), genoteerd met $L\theta$;*

Definitie 128 *LL-term Een literal-lijst-term (ll-term) is:*

- *Een lijst van l-termen is een ll-term, genoteerd met $\langle L_1, L_2, \ldots, L_n\rangle$. Het aantal l-termen kan niet oneindig zijn;*
- *Een literal-lijst-variabele is ll-term;*
- *Een ll-term gevolgd door een substitutie is een ll-term.*

L-termen en ll-termen zijn voorbeelden van logische struktuur-termen. Door het aantal termen uit te breiden kan een algemene logische struktuur-logika worden gekomponeerd.

---

[1] Voor implikatie wordt $\rightarrow$ gehanteerd in plaats van het meer gebruikelijke $\supset$ om verwarring met deelverzamelingen te voorkomen.

Definitie 129 *Wff Met behulp van de zojuist gedefinieerde termen kunnen de wff worden gedefinieerd:*

- *Een ll-term is een wff;*
- *Als α en β l-termen zijn, dan is $(\alpha = \beta)$ een wff;*
- *Als α en β ll-termen zijn, en $=^\phi$ een identiteitsrelatie, dan is $(\alpha =^\phi \beta)$ een wff;*
- *Konnektieven, kwantoren, leestekens en modale operatoren als gebruikelijk.*[2]

In een wff wordt een relatie tussen literal-lijsten weergegeven, niet tussen literals. Toch zijn literals (l-termen) ook gedefinieerd zodat er ook uitspraken over de literals van een literal-lijst kunnen worden gedaan. Over twee literal-lijsten die met dezelfde literal beginnen kan worden gesproken met behulp van:

(A.1) $\langle L_1, \ldots, L_n \rangle \wedge \langle L_{n+1}, \ldots, L_{2n} \rangle \wedge (L_1 = L_{n+1}) \rightarrow \ldots$

waarbij op de plaats van de laatste ... de gevolgen van deze gelijkheid kunnen worden weergegeven.

In deze meta-logika wordt niet gesproken over inverse substituties, zodat hier niet het probleem speelt deze korrekt te definiëren, hetgeen wel moet in een induktieve logika. Het draait hier slechts om afleidbaarheid, niet om efficiëntie.

Als een clause C_1 een inverse substitutie is van een andere clause C_2, wordt hier volstaan met de opmerking dat C_2 een substitutie is van C_1. Voor een uitwerking kan gebruikt worden gemaakt van bijvoorbeeld termpartities.

Er wordt in het volgende uitgegaan van universele afsluitingen van wff, zodat de formules korter en overzichtelijker zijn.

Afleidingsregels en axiomata

Als afleidingsregels gelden de gebruikelijke modus ponens en generalisatie. Merk op dat generalisatie geldt voor de metavariabelen (dus literal-variabelen, literal-lijst-variabelen en substitutie-variabelen), niet voor de variabelen die voorkomen in de literals: in predikatenlogika geldt dat als $P(x)$ afleidbaar (geldig) is, ook $P(a)$ afleidbaar is, dit is niet zo in de meta-logika! Literals en literal-lijsten zijn konstanten.

Als axiomata gelden de standaard axiomata. De definities van deze axiomata geven de gebruikelijke logische herschikkingen van de konnektieven, als bijvoorbeeld $C_1 \vee C_2$ geldig is, is ook $C_2 \vee C_1$ geldig.

Naast de standaard-axiomata worden axiomata voor substituties en de modale operatoren gedefinieerd.

[2]Merk op dat variabelen bij kwantoren zowel literal-variabelen, literal-lijst-variabelen als substitutie-variabelen kunnen zijn.

Definitie 130 *Substitutie-axiomata Als C_2 een substitutie θ van C_1 is, dan geldt*[3]*:*

(A.2) $C_1\theta \rightarrow C_2$

(A.3) $C_2 \rightarrow C_1\theta$

Bovendien gelden voor alle modale operatoren die worden geïntroduceerd bepaalde axiomata. Eerst zullen de identiteitsrelaties $=^{\phi}$ worden gedefinieerd. De indentiteitsrelaties geven van twee literal-lijsten aan of zij behoren tot de volgende verzamelingen:

- literal en literal-lijst: syntaktisch identiek;
- clausal form: alfabetische variatie;
- permutatie: permutatie en alfabetische variatie;
- clause: seteigenschap, permutatie en alfabetische variatie;
- reduktie: reduceren, permutatie en alfabetische variatie;

Als twee literal-lijsten tot dezelfde clause behoren dan zijn zij afleidbaar door dubbele literals te schrappen of te introduceren, de literals door elkaar te husselen en alfabetische variatie te nemen.

Eerst wordt alfabetische variatie gedefinieerd. Een alfabetische variatie is een substitutie, en precies die substituties waarvan de inverse weer een substitutie is (die inverse is θ_1):

Definitie 131 *Alfabetische variatie Een substitutie θ is een alfabetische variatie, geschreven als θ^{av}, als geldt $\exists\theta_1(C = C\theta^{av}\theta_1)$.*

De identiteitsrelaties worden door middel van de volgende axiomata gedefinieerd:

- Literal-lijst: $(C = C)$;
- Clausal-form: $(C =^{cf} C\theta^{av})$.
- Permutatie: $(\langle L_1, \ldots, L_i, \ldots, L_n\rangle =^{p} \langle L_i, L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_n\rangle\theta^{av})$.

  Door te stellen dat een willekeurige literal naar voren gehaald mag worden, zijn alle permutaties afleidbaar. Als bij n literals achteraan wordt begonnen en rekursief de vorige steeds naar voren wordt gehaald, dan is na n iteraties de gezochte permutatie gevonden[4].
- Clause:

[3] Merk op dat dit nog niet zo mooi geformuleerd is. Wellicht is het handiger deze als modale operatoren te definiëren.

[4] Een oneindige lijst wordt pas na oneindig veel iteraties bereikt, een reden om geen oneindige lijsten toe te staan zoals hier is gedaan (en gebruikelijk is binnen de clause logika).

- Permutatie: $(\langle L_1, \ldots, L_i, \ldots, L_n\rangle =^{cl} \langle L_i, L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_n\rangle\theta^{av})$.
- Reduktie van literals: $(\langle L_1, L_1, \ldots, L_n\rangle =^{cl} \langle L_1, \ldots, L_n\rangle$
- Expand literals: $(\langle L_1, \ldots, L_n\rangle =^{cl} \langle L_1, L_1, \ldots, L_n\rangle)$

Als een lijst een wff is waarin identieke literals voorkomen, dan mag een dubbele worden verwijderd. Identieke literals mogen worden geïntroduceerd.

- Reduceerbaar:
  - Permutatie: $(\langle L_1, \ldots, L_i, \ldots, L_{i-1}, L_{i+1}, \ldots, L_n\rangle =^{r} \langle L_i, L_1, \ldots, L_n\rangle\theta^{av})$.
  - Reduktie van literals: $(\langle L_1, L_2, \ldots, L_n\rangle\theta =^{r} \langle L_2, L_2, \ldots, L_n\rangle) \rightarrow (\langle L_1, L_2, \ldots, L_n\rangle =^{r} \langle L_2, \ldots, L_n\rangle\theta)$
  - Expand van literals: $(\langle L_1, L_2, \ldots, L_n\rangle\theta =^{r} \langle L_2, L_2, \ldots, L_n\rangle) \rightarrow (\langle L_2, \ldots, L_n\rangle\theta =^{r} \langle L_1, L_2, \ldots, L_n\rangle)$

  Dit is de ekwivalentie-klasse onder subsumptie, oftewel de reduktie-klasse. Merk op dat de clause-axiomata worden geïmpliceerd door de korresponderende axiomata. Deze definitie geeft ook de basis voor een inverse reduktie-algoritme.

  Van de literal-lijst $\langle P(a)\rangle$ kan nu ook $\langle P(a), P(a)\rangle$ en $\langle P(x), P(a)\rangle$ worden afgeleid (en andersom).

Op deze wijze is op een alternatieve wijze gedefinieerd wat clausal forms, clausal form permutaties, clauses en gereduceerde clauses zijn, namelijk verzamelingen van literal-lijsten. Merk op dat de verzameling van een gereduceerde clause bij deze definitie een superset is van de verzameling clauses, die op zijn beurt weer een superset is van de verzameling van een clausal form permutatie, die op zijn beurt weer een superset is van de verzameling clausal forms.

Deze verzamelingen-interpretatie maakt het mogelijk een alternatieve definitie te kiezen, die van kategorie-theorie. In deze formulering is echter eksakt bekend wat tot de verzameling behoort, wat later zijn voordelen zal hebben. Als intuïtief hulpmiddel zal de kategorie-theorie wel worden gehanteerd.

Semantiek

De semantiek is de gebruikelijke Kripke-semantiek voor modale logika. Iedere wereld heeft een aantal wff die waar zijn, in een wereld gelden Herbrand interpretaties en modellen. Modale operatoren definiëren werelden die wff geven met bepaalde eigenschappen ten opzichte van de huidige wereld.

A.2.2 Basisverzameling

Definitie 132 *Basisverzameling De huidige basisverzameling is de verzameling van grond ll-termen zonder substituties die geldig zijn. De basisverzameling van een modale operator is de verzameling van grondtermen zonder substituties die, voorafgegaan door de modale operator, geldig zijn.*

De basisverzameling is dus een verzameling van literal-lijsten. Als twee basisverzamelingen worden vergeleken kan worden gesproken over sub- en supersets.

Met behulp van de identiteitsregels kunnen een aantal axiomata worden gedefinieerd, die garanderen dat de basisverzameling altijd clausal forms, permutaties, clauses of redukties zijn. Als deze axiomata niet worden gedefinieerd maar wel geldig zijn, dan weten we dat het clausal forms, permutaties etc. zijn.

Van bijvoorbeeld de inverse substituties van een clause weten we dat deze substituties ook weer clauses zijn. We geven aan dat dit axioma geldig is – gedefinieerd of afleidbaar – door in de ll-termen een superscript te plaatsen. Dit wederom om de leesbaarheid te verhogen door de formules zo kort mogelijk te houden.

- Clausal-form: $C_1 \wedge (C_1 =^{cf} C_2) \rightarrow C_2$;

  LL-termen in een basisverzameling waar dit axioma geldig is, wordt genoteerd met C^{cf}.

- Permutatie: $C_1 \wedge (C_1 =^{p} C_2) \rightarrow C_2$;

  LL-termen in een basisverzameling waar dit axioma geldig is, wordt genoteerd met C^{p}.

- Clause: $C_1 \wedge (C_1 =^{cl} C_2) \rightarrow C_2$;

  LL-termen in een basisverzameling waar dit axioma geldig is, wordt genoteerd met C^{cl}.

- Reduktie: $C_1 \wedge (C_1 =^{r} C_2) \rightarrow C_2$;

  LL-termen in een basisverzameling waar dit axioma geldig is, wordt genoteerd met C^{r}.

A.2.3 Relaties tussen verzamelingen

Als eerste unaire modale operator wordt de identiteitsrelatie gedefinieerd: als een literal-lijst C geldig is, dan is ook $O_{id}C$ geldig. Daar in de modale wereld andere axiomata kunnen gelden, is dit toch interessant.

Definitie 133 *Id operator* $C \rightarrow O_{id}C$

Stel bijvoorbeeld dat $\langle P(x), Q(y)\rangle$ geldig is. Dan betekent dat via dit axioma dat $O_{id}\langle P(x), Q(y)\rangle$ ook geldig is. Lijkt niet erg indrukwekkend, maar stel nu dat in de modale wereld de clause-axiomata geldig zijn. Dan is ook $O_{id}\langle Q(y), P(x)\rangle$ geldig!

De volgende uitspraken zijn geldig:

- $O_{id}C \rightarrow C^{cf}$;
- $O_{id}C^{cf} \rightarrow C^{p}$;
- $O_{id}C^{p} \rightarrow C^{cl}$;

- $O_{id}C^{cl} \to C^r$;

Merk op dat C^{cf} en C^p dezelfde variabele is, alleen gelden er andere axiomata. De tweede regel is bijvoorbeeld een verkorte schrijfwijze voor: voor iedere literal-lijst C geldt dat alle literal-lijsten die met behulp van het clausal-form-axioma kunnen worden afgeleid, ook met behulp van het clausal-form-axioma en het permutatie-axioma kan worden afgeleid (triviaal).

Als er in de modale wereld andere axiomata gelden, dan kan bijvoorbeeld bepaald worden welke clauses worden geïnduceerd door clausal forms: $C_1^{cf} \to O_{id}C_2^{cl}$ geeft voor een clausal form C_1 alle clauses C_2.

Laat $C^{cf} = \langle P(y), P(x), Q(y)\rangle^{cf}$ een wff zijn. Dan zijn ook wff:

- Via het clausal form axioma: $\langle P(z), P(x), Q(z)\rangle^{cf}$, $\langle P(z), P(y), Q(z)\rangle^{cf}$, ...
- Via de zojuist gegeven definitie naar C^r[5]: $O_{id}\langle P(z), P(x), Q(z)\rangle^r$, $O_{id}\langle P(z), P(y), Q(z)\rangle^r$, ...
- Via het permutatie axioma: $O_{id}\langle P(x), Q(z), P(z)\rangle^r$
- Via het reduktie axioma: $O_{id}\langle Q(z), P(z)\rangle^r$

A.2.4 Monotone operatoren

Definitie 134 *Monotoon Een operator O is monotoon als geldt:*

(A.4) $(C_1 \to OC_2) \to (C_1 \wedge C_3 \to OC_2)$

Voor monotone operatoren geldt dat als iets voor een aantal wff geldt, dit ook geldt voor een superset van wff. Ofwel, als een operator geldt voor clausal forms, dan zijn deze clausal forms voorbeelden van permutaties etc.

De operatoren hebben soms axiomata van de vorm $p_1 \vee \ldots \vee p_n \to OC$, waarbij de p_i eisen zijn waaraan een literal-lijst moet voldoen om een O genoemd te mogen worden. Als alle p_i positief zijn, is de operator monotoon.

Monotone operatoren hebben de plezierige eigenschap dat als de basisverzameling uit meerdere verzamelingen bestaat, in de modale wereld iedere verzameling geldig is die geldig zou zijn als één van de basisverzamelingen geldig zou zijn. Vaak is dit niet zo, daarom wordt gedefinieerd:

Definitie 135 *Relatief monotoon Een operator O is monotoon, relatief aan de identiteitsrelatie $=^\phi$, als geldt:*

(A.5) $((C_1 \to OC_2) \wedge (C_1 =^\phi C_3)) \to (C_1 \wedge C_3 \to OC_2)$

[5] Het is nu niet meer duidelijk dat de reduktie-axiomata geldt op O_{id}. In feite is er voor ieder axioma een andere modale operator. Daar we meestal toch over slechts één soort praten is de introduktie van verschillende operatoren niet echt zinvol. Alternatief is definiëren van geldende axioma's op de operator, dus O_{id}^r

Straks bij de strikte generalisaties zal het nut hiervan blijken. Overigens is iedere relatief monotone modale operator om te schrijven naar monotone operatoren door ekstra eisen in te bouwen. Deze monotone operatoren gelden dan nog slechts voor bepaalde verzamelingen!

A.3 Enige modale operatoren

In deze paragraaf wordt de meta-logika toegepast. Er wordt gekeken naar bekende relaties tussen twee clausal forms c.q. clauses, namelijk generalisatie, subsumptie, strikte generalisatie en cover.

A.3.1 Generalisatie

Definitie 136 *Generalisatie Generalisatie wordt als inverse substitutie gedefinieerd:*

(A.6) $\forall C \forall \theta (C\theta \rightarrow O_{is}C)$

Dit is de inverse van substitutie:

(A.7) $\forall C \forall \theta (C \rightarrow O_s C\theta)$

Dit is monotoon, dus: een generalisatie van clausal forms induceert een generalisatie van korresponderende clauses en een generalisatie van clauses induceert een generalisatie voor alle clausal forms.

(A.8) $(C_1^{cf} \rightarrow O_{is}C_2^{cf}) \rightarrow (O_{id}C_1^{cl} \rightarrow O_{is}O_{id}C_2^{cl})$

(A.9) $\exists C_2(C_1^{cl} \rightarrow O_{is}C_2^{cl}) \rightarrow (O_{id}C_1^{cf} \rightarrow O_{is}O_{id}C_2^{cf})$

A.3.2 Subsumptie

Het laten vallen van literals, uit axioma $p \rightarrow p \vee q$:

Definitie 137 *Drop Drop:*

(A.10) $\forall (L_1, \ldots, L_n)\langle L_1, \ldots, L_i, \ldots, L_n\rangle \rightarrow O_d\langle L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_n\rangle$

Definitie 138 *Subsumptie Subsumptie:*

(A.11) $\forall C(O_{is}C \vee O_{is}O_dC) \rightarrow O_{sub}C$

De drop- en subsumptie-operator zijn monotoon.

A.3.3 Strikte generalisatie

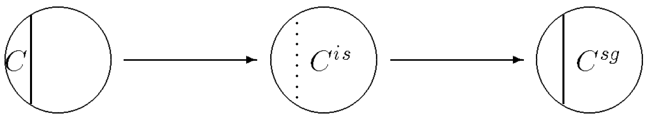
Interessanter wordt het als strikte generalisaties worden bekeken, dit is namelijk niet monotoon. Hierin komt bij de voorwaarden een eis voor over de conclusie, namelijk: er mag geen literal-lijst bestaan zdd ... Dit betekent dat deelverzamelingen nog geen eisen voor supersets garanderen. Eerst definiëren we strikte generalisatie, ervan uitgaande dat de basisverzameling uit één verzameling bestaat.

Definitie 139 *Strikte generalisatie Strikte generalisatie als:*

(A.12) $\forall C(O_{is}C \wedge \neg C) \rightarrow O_{sg}C$

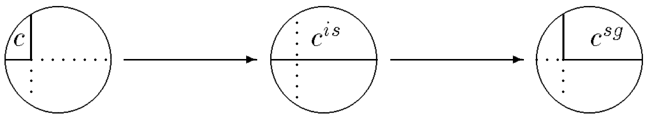
De eerste eis betekent dat C een inverse substitutie is, de tweede dat C niet afleidbaar is. Deze tweede eis is een beperking van het aantal generalisaties dat afgeleid kan worden, dus een superset kan niet meer garanderen dat iedere subset ook een subset impliceert.

Er wordt geanalyseerd of bepaalde subsets supersets impliceren, zoals het volgende plaatje illustreert. De cirkel stelt de verzameling voor van alle inverse substituties C^{is}, de literal-lijsten C vormen een strikte deelverzameling hiervan en de strikte generalisaties C^{sg} zijn precies die inverse substituties die geen literal-lijsten zijn:



Voor een subset c, geïnduceerd door de superset C, geldt $c \subseteq C$ en $c^{is} \subseteq C^{is}$ (want is monotoon). De monotoon-eis is dat ook $c^{sg} \subseteq C^{sg}$. Dit geldt dus niet voor iedere willekeurige sub- en superset, want strikte generalisatie is niet monotoon.

Dit is ekwivalent met de eis dat de doorsnede van C en c^{sg} leeg is, of dat de subset c precies de doorsnede van c^{is} en C is.



Stelling 38 *Strikte generalisatie voor permutaties Strikte generalisatie is monotoon relatief aan $=^p$.*

Bewijs 28 *Zie paragraaf 5.3.1*

Er is dus een alternatieve definitie voor strikte generalisatie, die monotoon is voor permutaties, wat wil zeggen dat als er meerdere permutaties in de basisverzameling zitten, de modale operator toch alle generalisaties geeft[6].

Definitie 140 *Strikte generalisatie voor permutaties Strikte generalisatie voor permutaties als:*

(A.13) $(C_1 \wedge (C_1 \rightarrow O_{is}C_2) \wedge \neg(C_1 =^{p} C_2)) \rightarrow O_{sg}C_2$

En er geldt:

(A.14) $(C_1^{cf} \rightarrow O_{sg}C_2^{cf}) \rightarrow (O_{id}C_1^{p} \rightarrow O_{sg}O_{id}C_2^{p})$

(A.15) $\exists C_2(C_1^{p} \rightarrow O_{sg}C_2^{p}) \rightarrow (O_{id}C_1^{cf} \rightarrow O_{sg}O_{id}C_2^{cf})$

Stelling 39 *Strikte generalisatie voor clauses Strikte generalisatie is monotoon relatief aan clauses.*

Bewijs 29 *Zie paragraaf 5.4.2.*

En voor set-generalisatie geldt:

Definitie 141 *Strikte generalisatie voor clauses Strikte generalisatie voor clauses als:*

(A.16) $(C_1 \wedge (C_1 \rightarrow O_{is}C_2) \wedge \neg(C_1 =^{cl} C_2) \rightarrow O_{sg}C_2$

(A.17) $(C_1^{p} \rightarrow O_{sg}C_2^{p}) \rightarrow (O_{id}C_1^{cl} \rightarrow O_{sg}O_{id}C_2^{cl})$

(A.18) $\exists C_2(C_1^{cl} \rightarrow O_{sg}C_2^{cl}) \rightarrow (O_{id}C_1^{p} \rightarrow O_{sg}O_{id}C_2^{p})$

Stelling 40 *Strikte generalisatie voor redukties Strikte generalisatie is niet monotoon relatief aan redukties.*

Bewijs 30 *Zie paragraaf 5.5.1.*

A.3.4 Covers

Strikte generalisatie is monotoon, relatief aan clauses, permutaties en clausal forms. Dit betekent dat een strikte generalisatie van clausal forms ook een strikte generalisatie is van een clause waartoe de clausal form behoort en dat de strikte generalisatie van een clause gelijk is aan de som van de strikte generalisaties van zijn clausal forms.

En het betekent dat in de aangepaste definities monotoon zijn (voor permutaties en clauses), dus dat als enkele clauses geldig deze definitie de vereniging van strikte generalisaties van de clauses geldig maakt.

[6] Dit ziet er niet zo best uit. Misschien moeten alle modale operatoren als binaire operatoren gedefinieerd worden, maar het voordeel van het gebruik van een meta-logika lijkt dan verloren te gaan.

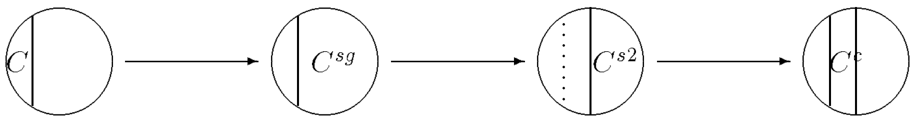
De strikte generalisatie van strikte generalisaties, dat is een ander probleem. Door de niet-modulaire struktuur van de tralie van literal-lijsten is dit niet triviaal. Neem bijvoorbeeld $\langle P(f(x), f(x))\rangle$ dan zijn enkele strikte generalisaties ervan $\langle P(x, x)\rangle$, $\langle P(x, y)\rangle$, $\langle P(f(x), f(y))\rangle$, $\langle P(f(x), y)\rangle$. Er is dus een keten van covers $P(f(x), f(x)) < P(x, x) < P(x, y)$ en een chain $P(f(x), f(x)) < P(f(x), f(y)) < P(f(x), y) < P(x, y)$. Covers van covers is dus niet triviaal, covers zelf wel?

Een cover is een strikte generalisatie zonder kleinere generalisatie. Dus alle literal-lijsten die een strikte generalisatie zijn maar niet een strikte generalisatie van een strikte generalisatie. Voor monotone operatoren van strikte generalisatie wordt gedefinieerd[7]:

Definitie 142 *Cover Cover:*

$$\text{(A.19)} \quad \forall C(O_{sg}C \wedge \neg O_{sg}O_{sg}C) \rightarrow O_c C$$

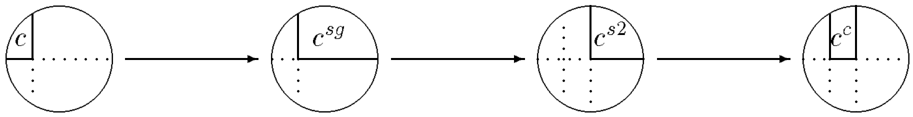
Er wordt geanalyseerd of bepaalde subsets supersets impliceren, zie wederom het volgende plaatje. De cirkel stelt de verzameling voor van alle inverse substituties C'^{is}, de literal-lijsten C vormen een strikte deelverzameling hiervan en de strikte generalisaties C'^{sg} zijn precies die inverse substituties die geen literal-lijsten zijn. De strikte generalisaties van de strikte generalisatie C'^{s2} zijn wederom een deelverzameling van de strikte generalisaties C'^{sg}, en het verschil tussen de strikte generalisaties van de strikte generalisaties en de strikte generalisaties zijn de covers:



Voor een subset c, geïnduceerd door de superset C, geldt $c \subseteq C$ en $c^{is} \subseteq C'^{is}$ (want is monotoon). Voor clausal forms, permutaties en clauses, maar niet voor gereduceerde clauses, geldt dat $c^{sg} \subseteq C'^{sg}$. De monotoon-eis is dat ook $c^c \subseteq C'^c$. Dit geldt dus niet voor iedere willekeurige sub- en superset, want cover is niet monotoon.

Dit is equivalent met de eis dat de doorsnede van C'^c en c^{s2} leeg is, of dat de subset c^c precies de doorsnede van c^{sg} en C'^c is.

[7] Als strikte generalisatie niet monotoon is, gaat het fout; dus de alternatieve definities van de vorige paragraaf.



Stelling 41 *Covers Covers zijn monotoon, relatief aan clausal forms en permutaties maar niet relatief aan clauses en gereduceerde clauses.*

Bewijs 31 *Zie paragraaf 5.2, 5.3.2 en 5.4.3.*

A.4 Logische strukturen

De in de vorige paragraaf gedefinieerde meta-logika is gehanteerd voor analyse van de verschillende kwasi-ordeningen zonder achtergrondtheorie. De logika der logische strukturen zou ook kwasi-ordeningen met achtergrondtheorie aan moeten kunnen, en uiteindelijk is het ideaal de meta-logika geschikt te maken voor analyse van induktieve algoritmen.

In deze laatste paragraaf worden de mogelijke uitbreidingen besproken en wordt gedroomd over mogelijke toepassingen.

A.4.1 Syntaks

De in de vorige paragraaf gedefinieerde meta-logika heeft als termen literals en lijsten van literals (die clausal forms representeren). Als ook over logische theorieën geredeneerd moet kunnen worden, moeten de termen ook deze logische theorieën kunnen representeren, ofwel er ontstaan literal-lijst-lijsten.

Als vervolgens ook algoritmen met behulp van deze meta-logika worden geanalyseerd, moeten ook achtergrondtheorie en (positieve en negatieve) voorbeelden kunnen worden opgenomen in de termen. Zo ontstaan komplekse lijststrukturen, de zogenaamde logische strukturen.

Er zullen bovendien een aantal ekstra modale operatoren geïntroduceerd moeten worden. Hierbij valt te denken aan suprema van clauses en de V- en W-operator, met alle restrikties hierop die in de literatuur zijn voorgesteld.

De V-operator is vervolgens de basis voor inverse lineaire afleidingen (de funktie V^n) en suprema van deze inverse lineaire afleidingen. Deze samengestelde operatoren zijn een handvat bij de analyse of relatieve subsumptie en gegeneraliseerde subsumptie haalbaar zijn met als basis de V-operator (en subsumptie).

Naast deze *bottom-up* benadering kunnen vervolgens ook de *top-down* operatoren (zoals de *refinement operatoren*) in de logika worden opgenomen.

A.4.2 Klassificeren operatoren

Bij de analyse van de verschillende (versies van) operatoren kan gebruik worden gemaakt van abstrakte eigenschappen van de operatoren, zoals in de vorige paragraaf gebruik werd gemaakt van de monotoniteit-eigenschap.

In de modale logika is een dergelijke klassifikatie gebruikelijk en systemen die aan bepaalde eisen voldoen hebben bekende namen [18]. Hierbij kan worden gedacht aan refleksiviteit (`T`), transitiviteit (`S4`), symmetrie (`B`) etc.

A.4.3 Kombineren operatoren

Een algoritme bestaat uit een aantal operatoren die in een bepaalde volgorde worden aangeroepen. Interessant wordt de meta-logika dus pas als de operatoren kunnen worden gekombineerd. Hierbij kan gedacht worden aan het sekwentieel uitvoeren van operatoren en keuzen tussen operatoren. Een opvolging van operatoren is in de vorige paragraaf getoond bij de strikte generalisatie van strikte generalisatie en keuzen bij de subsumptie-operator.

Door wat te eksperimenteren met deze meta-logika werd duidelijk dat aan de kwaliteit van voorbeelden, die worden ingevoerd in de huidige lerende systemen, een aantal eisen wordt opgelegd waaraan in werkelijkheid waarschijnlijk niet altijd voldaan zal zijn. Dit wordt veroorzaakt door twee problemen van de kombinatie van V-operator met subsumptie: verschilgevoeligheid en bodemgevoeligheid.

Het eerste probleem, de verschilgevoeligheid, geeft aan dat twee voorbeelden veel op elkaar moeten lijken om bruikbaar te zijn voor induktieve operatoren. Een voorbeeld hiervan, van Lapointe en Matwin [29], is in paragraaf 6.4.4 behandeld.

Het tweede probleem, de bodemgevoeligheid, ligt ook bij rekursieve clauses: als de bodem van de rekursie niet bekend is, zal de induktieve redenering zeer inefficiënt zijn en de korrekte clause waarschijnlijk niet worden afgeleid. Dit probleem speelt ook bij het systeem van Lapointe en Matwin [29].

A.4.4 Efficiëntie

De meta-logika, zoals tot nu toe besproken, is een analyseapparaat voor afleidbaarheid van clauses. Afleidbaarheid is echter maar één kant van de medaille; aan de andere kant staat efficiëntie. Des te meer een operator af kan leiden, des te inefficiënter hij wordt.

Het zou mooi zijn als de meta-logika ook iets zinnigs over de efficiëntie zou kunnen zeggen. De eenvoudigste oplossing zou zijn het aantal (modale) operatoren – het aantal stappen – te tellen dat afgelegd moet zijn voor een bepaalde clause of verzameling clauses afgeleid is.

Dit is echter slechts een zeer grove indikatie; de beste oplossing zou zijn als er een tijdmeting gedaan zou kunnen worden. Daarvoor zou de meta-logika uitvoerbaar moeten zijn.

A.4.5 Programma

De droom bereikt nu zijn hoogtepunt. Als aan de meta-logika een efficiënte deduktietheorie wordt toegevoegd, dan kan de meta-logika niet alleen als analysemethode gebruikt worden maar ook als efficiëntiemeter en uiteindelijk als *workbench* bij de konstruktie van leeralgoritmen.

Uitvoerbaarheid was echter niet het doel bij het opzetten van de meta-logika, en in haar huidige is zij daar ook niet erg geschikt voor. Dit is vergelijkbaar met de twee toepassingen van logika binnen filosofie en informatika; van de filosofie-methode moet worden overgestapt op de informatika-methode.

Het huidige systeem is gebaseerd op een modale predikatenlogika en zal beperkt moeten worden tot Prolog. Deze uitbreiding van Prolog wordt gewoonlijk gedaan door gebruik te maken van de CWA, ik betwijfel echter of dat hier zou werken. De modale operatoren kunnen worden herschreven naar predikaten maar een andere oplossing (een deontologisch systeem) zou de voorkeur verdienen.

De centrale gedachte is: *de beschrijving is ook het programma.* Dit lijkt sensationeel, maar het is geen nieuwe gedachte binnen de informatika.

Een voorbeeld hiervan is een grafische gebruikersinterface, waarbij een grafische representatie (de beschrijving) wordt vertaald naar een programma. Lee [25] beschrijft een systeem, `CASE-EDI`, waarin een grafische representatie in de vorm van Petri-netten wordt omgezet naar een Prolog-programma.

Een ander voorbeeld is het systeem van Gommans en Nales [16], waarin de beschrijving van de *administratieve organisatie* (AO, een methode van accountants om organisaties te analyseren) wordt gebruikt als basis voor een informatiesysteem van bedrijven.

Het slot van de droom:

De laatste sessie, waarin AO en AI botsen en sommige mensen spijt hebben.

Plaats	: kantoor in een wolkenkrabber '*on the other side of the ocean*'.		
Aanwezig	: Dr `G`, dr `N`	:	direkteuren.
	`T`	:	systeemanalist.
	`A`	:	accountant.

`A`: Heren, u zit in een krisisituatie: uw bedrijf staat op het punt bankroet te gaan. Ik heb uw werknemers op een monitor van het informatiesysteem gevolgd en het blijkt dat secretaresse S een aantal kritieke verkooporders heeft gegeven.

`N`: Oops, hoe kon zij die beslissingen nemen?

`T`: Ja, ik denk dat ik dat wel kan verklaren. Zoals u weet zijn wij reeds enige tijd aan het werk het informatiesysteem van intelligentie te voorzien. De nieuwe lerende module, waar ik u onlangs een memo over heb gestuurd, heeft blijkbaar enige wijzigingen in de funktiedefinities aangebracht om de funktiescheiding korrekt door te voeren.

N: Mijnheer T, vindt u die wijzigingen intelligent?

T: Een interessante vraag. Ik denk dat intelligentie, om Turing te citeren, een ...

N: Maar het is uw schuld dat wij failliet gaan!

T: Ik weet dat u uw titel in de Administratieve Informatika heeft behaald, maar U moet toch begrijpen dat het leren binnen de *Artificial Intelligence* uiterst kompleks is. Wat die schuld betreft, ik denk dat wij allen ...

A: Heren, ik denk dat u beter niet met deze man in zee had kunnen gaan.

G: Ik heb die intelligente informatiesystemen nooit zo'n goed idee gevonden.

Referenties

[1] Ranan B. Banerji. *Learning Theories in a subset of a Plyadic Logic.*

[2] Henk Barendregt, *The Incompleteness Theorems.* Rijksuniversiteit Utrecht, 1976.

[3] J.C. Bioch & P.R.J. van der Laag. *Simple Improvements of a Simple Solution for Inverting Resolution.* Report EUR-CS-91-03. Erasmus Universiteit, Rotterdam, maart 1991.

[4] Lenore Blum & Manuel Blum. Toward a Mathematical Theory of Inductive Inferences. In: *Information and Control* 28. Academic Press, 1975.

[5] Margaret A. Boden. *The Philosophy of Artificial Intelligence.* Oxford University Press, 1990.

[6] J.D. Bolter. *Turing's Man.* North Carolina Press.

[7] Alain Bonnet. *Kunstmatige Intelligentie: Verwachtingen en Werkelijkheid.* Addison-Wesley, Amsterdam, 1984.

[8] L. G. Bouma. Formele Achtergronden van PROLOG. In: *Informatie* 29(4), pag. 316 - 327, 1987.

[9] Wray Buntine. Generalized Subsumption and its Applications to Induction and Redundancy. In: *Artificial Intelligence* 36(2), pag. 149 - 176, 1988.

[10] H. Coenders (ed). *Kramers Nieuw Woordenboek.* Argus, Amsterdam, 1990.

[11] Peter A. Flach. A Framework for Inductive Logic Programming. In: S.H. Muggleton(Ed). *Proceedings Inductive Logic Programming.* Academic Press, 1991.

[12] Peter A. Flach. *The Logic of Explanations.*

[13] Michael R. Genesereth & Nils J. Nilsson. *Logical Foundations of Artificial Intelligence.* Palo Alto, 1987.

[14] A. Giordana, L. Saitta & D. Roverso. *Abstracting Concepts with Inverse Resolution.*

[15] A. Giordana, L. Saitta, R. Finelli, M. Paderni & D. Roverso. *Extending Inverse Resolution to Build up Abstraction.*

[16] L.F.B. Gommans & A. Nales. *Modelleren van de Administratieve Organisatie.* Erasmus Universiteit, Rotterdam, 1992.

[17] Douglas R. Hofstadter. *Gödel, Escher, Bach: een Eeuwige Gouden Band.* Contact, Amsterdam, 1985.

[18] G.E. Hughes & M.J. Cresswell. *A Companion to Modal Logic.* Methuen & Co, Londen, 1984.

[19] David Hume & Claude Sammut. *Using Inverse Resolution to Learn Relations from Experiments.*

[20] Hiroki Ishizaka. Model Inference Incorporating Generalisation. In: *Journal of Information Processing* 11(3). 1988.

[21] Willem R. de Jong. *Formele Logika, een Inleiding.* Coutinho, Muiderberg, 1988.

[22] Kimmerle (ed). *Syllabus Filosofie van de Twintigste Eeuw.* Erasmus Universiteit, Rotterdam, 1990.

[23] Patrick van der Laag. Een Meest Algemene Verfijningsoperator voor Gereduceerde Zinnen. In: *Proceedings of NAIC '92.*

[24] K. Lano & H. Haughton. Reasoning and Refinement in Object-Oriented Specification Languages. In: *Lecture Notes in Computer Science 615, Proceedings ECOOP '92*, pag. 78 - 97. Springer-Verlag, Berlijn, 1992.

[25] Ronals M. Lee. Dynamic Modelling of Documentary Procedures: a Case for EDI. In: *Proceedings of the Third International Working Conference on Dynamic Modelling of Information Systems.* Noordwijkerhout, 1992.

[26] Xiaofeng (Charles) Ling & Malur Aji Narayan. *A Critical Comparison of Various Methods Based on Inverse Resolution.*

[27] J.W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, Berlijn, 1984.

[28] M.J. Maher. Equivalences of Logic Programs. In: *Proceedings of Third International Conference on Logic Programming.* Springer, Berlijn, 1986.

[29] Matwin & Lapointe. *Sub-unification: a Tool for Efficient Induction of Recursive Programs.* Ottowa University Press, Ottowa, jan. 1992.

[30] T.M. Mitchell. Generalization as Search. In: *Artificial Intelligence* 18(2), pag. 204 - 226. 1982.

[31] Stephan Muggleton & Wray Buntine. Machine Invention of First-order Predicates by Inverting Resolution. In: *Proceedings of the Fifth International Conference on Machine Learning*, pag. 339 - 352. Morgan Kaufmann, San Mateo, 1988.

[32] Stephen Muggleton. Inductive Logic Programming. In: *Proceedings of the First Conference on Algorithmic Learning Theory.* Ohmsha, Tokyo, Japan, 1990.

[33] Stephen Muggleton, Ashwin Srinivasan & Micheal Bain. *MDL Codes for Non-Monotonic Learning.* Turing Institute Technical Report, 1991.

[34] R.P. Nederpelt. *De Taal van de Wiskunde.* Versluys, Almere, 1987.

[35] Tim Niblett. A Study of Generalisation in Logic Programs. In: *European Working Sessions on Learning.* Pitman, Londen, 1988.

[36] Tim Niblett. *Plausible Inference and Negation in Horn Clause Logic.*

[37] Shan-Hwei Nienhuys-Cheng. *The V- and W-operators in Inverse Resolutions.*

[38] Shan-Hwei Nienhuys-Cheng & Peter A. Flach. Consistent Term Mappings, Term Partitions and Inverse Resolution. In: Y. Kodratoff (Ed). *European Working Session on Learning* 1, pag. 361 - 374. Springer Verlag 482, 1991.

[39] Shan-Hwei Nienhuys-Cheng. *Consequent Functions and Inverse Resolutions.* Report EUR-CS-90-03. Mei 1990.

[40] Shan-Hwei Nienhuys-Cheng. *Term Partitions and Minimal Generalizations of Clauses.* Report EUR-CS-91-01. Januari 1991.

[41] Shan-Hwei Nienhuys-Cheng. *A Survey of Generalization Algorithms and Inverse Resolutions.*

[42] Shan-Hwei Nienhuys-Cheng. *Codings Related to Flattening and Term Partitions.*

[43] Shan-Hwei Nienhuys-Cheng. *Flattening, Generalizations of Clauses and Absorption Algorithms.* Report EUR-CS-91-02. Maart 1991.

[44] Shan-Hwei Nienhuys-Cheng & L.W.N. van der Torre. *Permutation Generalizations, Term Partitions and Syntactic Generalizations.*

[45] Friedrich Nietzsche. *De Vrolijke Wetenschap.* Arbeiderspers, Amsterdam, 1976.

[46] Nijholt en Steels. *Ontwikkelingen in Expert-Systemen.* Academic service, Den Haag, 1986.

[47] Gordon D. Plotkin. A Note on Inductive Generalization. In: B. Meltzer, D.Michte(eds). *Machine intelligence* 5, Edinburgh University Press, Edinburgh, 1970.

[48] Gordon D. Plotkin. A Further Note on Inductive Generalization. In: B. Meltzer, D.Michte(eds). *Machine Intelligence* 6, Edinburgh University Press, Edinburgh, 1971.

[49] H. van Praag e.a.. *Kunstmatige Intelligentie.* Pandata, Rijswijk, 1988.

[50] Han Reichgelt. *Knowledge Representation; an AI Perspective.* Ablex publishing cooperation, Norwood, England, 1991.

[51] John C. Reynolds. Transformational Systems and the Algebraic Structure of Atomic Formulas. In: B. Meltzer, D. Michte(eds). *Machine intelligence* 5, Edinburgh university Press, Edinburgh, 1970.

[52] J.A. Robinson. A Machine-Oriented Logic Based on the Resolution Principle. In: *Journal of the Association for Computing Machinery*, 12(1), pag. 23-41, 1965.

[53] Céline Rouveirol & Jean-Francois Puget. A Simple Solution for Inverting Resolution. In: *EWSL-89*, pag 201 - 210. Pitman, Londen, 1989.

[54] Céline Rouveirol & Jean-Francois Puget. Beyond Inversion of Resolution. In: *Proceedings of the Fifth International Conference on Machine Learning.* Margan Kaufman, 1990.

[55] Céline Rouveirol. *Completeness for Inductive Procedures.*

[56] Ehud Y. Shapiro. Inductive Inference of Theories From Facts. In: *Technical Report 192.* Yale University, 1981.

[57] Ehud Y. Shapiro. *Algorithmic Program Debugging.* MIT press, 1983.

[58] Yoav Shoham. *Reasoning about change.* MIT Press, Cambridge, 1984.

[59] L. Sterling & E. Shapiro. *The Art of Prolog: Advanced Programming Techniques.* MIT-press, Cambridge, MA, 1986.

[60] Tarmo Uustalu. Combining Object-Oriented and Logic Paradigms: A Modal Logic Programming Approach. In: *Lecture Notes in Computer Science 615, Proceedings ECOOP '92*, pag. 98 - 113. Springer-Verlag, Berlijn, 1992.

[61] L. Vis. *Voortgangsverslagen Software Engineering.* Erasmus Universiteit, Rotterdam, 1990.

[62] Gerard de Vries. *De Ontwikkeling van Wetenschap, een Inleiding in de Wetenschapsfilosofie.* Wolters-Noordhoff, 2de druk, 1985.

[63] Ruediger Wirth. Completing Logic Programs by Inverse Resolution. In: *EWSL-89*, pag. 239 - 250. Pitman, Londen, 1989.

[64] Ludwig Wittgenstein. *Tractatus Logico-Philosophicus.* Polak & van Gennip, Amsterdam, 1989.

Index