Blok I

Inleiding

Inhoud hoofdstuk 1

Hoofdstuk 1

Inleiding

1.1 WAT IS LOGICA?

Logica is van origine de wetenschap die zich bezighoudt met de studie van *correct redeneren*. Reeds in de Oudheid werd opgemerkt dat menselijk redeneren systematisch valt te ontleden in combinaties van vaste patronen, zogenaamde *gevolgtrekkingen*. Voorbeelden van dergelijke gevolgtrekkingen zijn:

a Als je hoofdpijn hebt, dan lust je deze chocola niet.
Je lust deze chocola. Dus je hebt geen hoofdpijn.
b Alle kaaimannen zijn reptielen. Geen reptiel kan fluiten.
Dus geen kaaiman kan fluiten.

De verschillende kwesties die bij de systematische bestudering van zulke patronen vervolgens aan het licht traden, hebben geleid tot even zovele onderwerpen binnen de logica.

Om te beginnen, moeten we de *vorm* van gevolgtrekkingen zichtbaar maken. In natuurlijke taal (de 'gewone' taal) zit deze vorm vaak enigszins verstopt. Daarom ontwikkelden logici vele overzichtelijke notaties, die de betrokken beweringen weergeven in een structuur met variabele onderdelen en zekere vaste logische sleutelbegrippen. Vanuit dergelijke notatiesystemen hebben zich in de logica zogenaamde *formele talen* ontwikkeld, symbolische talen met een heel exacte opbouw, waarin de logische structuur van gevolgtrekkingen kan worden weergegeven. Een aantal van deze logische notaties zal in dit boek gaandeweg geïntroduceerd worden. Zo zal de structuur van de eerdergenoemde voorbeelden er in de volgende hoofdstukken als volgt komen uit te zien:

a	$h \rightarrow \neg c$	c	dus:	$\neg h$
b	$\forall x\,(Kx \rightarrow Rx)$	$\neg\exists x\,(Rx \wedge Fx)$	dus:	$\neg\exists x\,(Kx \wedge Fx)$

Het louter weergeven van gevolgtrekkingen is voor logici overigens slechts een eerste stap op weg naar het systematisch beschrijven van het gehele proces van *geldig* redeneren. Om dit te beschrijven ontwikkelde men systemen van *bewijzen* die, uitgaande van bepaalde geldige basisprincipes, door toegestane combinatieregels steeds complexere (gehelen van) gevolgtrekkingen kunnen produceren. Twee voorbeelden van dergelijke bewijssystemen, beide in wezen reeds uit de Oudheid afkomstig, zijn de *Boolese* (spreek uit: boelse) *logica* (waarmee voorbeeld a is te verantwoorden) en de klassieke *syllogistiek* (waaruit voorbeeld b stamt). Op deze en andere bewijssystemen komen we later uitgebreider terug.

Naast het idee dat een redenering geldig is indien zij te bewijzen valt volgens de spelregels van een logisch bewijssysteem, is er echter ook een andere invalshoek, die goed bij onze intuïtieve voorstellingen aansluit. We kunnen een gevolgtrekking ook *logisch geldig* noemen als de uitgangspunten een absolute garantie geven voor de conclusie, en wel in de volgende zin:

In elke situatie waarin de uitgangspunten waar zijn, is ook de conclusie waar.

Deze uitleg vooronderstelt een logische theorie over het begrip 'waarheid' en meer in het algemeen over de 'betekenis' van beweringen die in gevolgtrekkingen voorkomen. Er is dan ook een logische *semantiek* (betekenisleer) ontwikkeld die onderzoekt hoe formele talen precies aan hun betekenis komen. Deze betekenisleer is overigens ook toepasbaar gebleken buiten het gebied van geldige gevolgtrekkingen, zoals we nog zullen zien in de loop van dit boek. Met name heeft zij toepassingen in de studie van *natuurlijke talen*, zodat er een weg loopt van logica naar taalkunde, en tevens heeft zij model gestaan voor de semantiek van *programmeertalen*.

Ten slotte heeft de logica ook steeds een meer praktische interesse gehad in het effectief, mechanisch nagaan of van geldigheid sprake is in concrete gevallen. Daarmee komen we op de *algoritmiek* van zogenaamde 'beslissingsmethoden' om geldigheid voor gevolgtrekkingen in gegeven formele talen te testen: een droom die door de geschiedenis heen vele projecten voor logische 'redeneermachines' heeft geïnspireerd.

De historische ontwikkeling van de logica kent enkele hoogtepunten in de klassieke Oudheid (Aristoteles, 4e eeuw v. Chr.), de Middeleeuwen

(Scholastiek, ongeveer 1200) en het begin van de Moderne tijd (Leibniz, ongeveer 1700). Maar in een echte stroomversnelling raakte het vak pas in de vorige eeuw, toen wetenschappers ermee gingen werken binnen het *grondslagenonderzoek* van de wiskunde (George Boole, Gottlob Frege, David Hilbert). In de wiskunde bleek grote precisie geboden in het analyseren van redeneren. Deze toepassing bracht enkele standaardsystemen voort, zoals de *propositielogica* en de *predikaatlogica*.
Deze twee systemen staan in dit boek centraal. De propositielogica richt zich op de structuur van beweringen met zogenaamde *Boolese operatoren* (zie voorbeeld a; komt aan de orde in blok II van het boek). De predikaatlogica richt zich op beweringen met zogenaamde *kwantificerende uitdrukkingen* (zie voorbeeld b; blok III).

Toen deze standaardsystemen eenmaal ontworpen waren, bleek het ook mogelijk om diverse andere logische theorieën op dit stramien te ontwikkelen, die de genoemde twee aanvullen, inperken of er soms zelfs mee concurreren. In blok IV zullen we dergelijke variaties bespreken. Hoofdstuk 12 geeft een kort overzicht van zulke inperkingen van predikaatlogica als de equationele logica, uitbreidingen als hogere–orde logica en lambda–calculus, of alternatieven zoals de intuïtionistische logica. Een heel bekende uitbreiding van de standaardsystemen, oorspronkelijk ontwikkeld met het oog op logische toepassingen binnen de filosofie, is de zogenaamde *modale* logica, die haar werkterrein juist uitbreidt tot redeneren met begrippen die veranderende mogelijkheden in tijdsverloop, kennis of berekeningen beschrijven. Hoofdstuk 13 introduceert dit tegenwoordig veelgebruikte formalisme.

In dit boek wordt wat betreft de logica het volgende concrete doel beoogd: u vertrouwd te maken met de werking van de tot nog toe genoemde logische systemen. Maar op de achtergrond speelt ook een meer algemene bedoeling, te weten: u inzicht geven in logische begrippen en methoden *als zodanig*, die immers steeds ruimer toepasbaar zijn gebleken in de wiskunde en wijsbegeerte, gaandeweg ook in de taalkunde, en momenteel speciaal ook in de informatica. Over deze laatste ontwikkeling gaat het nu verder met name.

1.2 LOGICA EN INFORMATICA

Zoals reeds werd opgemerkt, behoort het bouwen van redeneermachines tot de voorgeschiedenis van 'mechanisering' van de logica. Pogingen daartoe werden reeds ondernomen in de Middeleeuwen. Een bekend voorbeeld uit de vorige eeuw is de machine van Charles

Babbage. De eerste meer moderne verwezenlijking van dit idee zijn de *Boolese schakelingen* (zie hoofdstuk 3). Meer in het algemeen leverde het grondslagenonderzoek van de wiskunde in de jaren dertig van deze eeuw een universeel begrip op van *effectieve berekenbaarheid*. Dit werd omschreven via abstracte zogenaamde Turing–machines, die later een inspiratiebron waren voor het ontwerpen van echte rekenautomaten in de jaren veertig (zie de inleiding tot blok V).

De belangrijkste hedendaagse verbanden tussen logica en informatica liggen echter niet op het gebied van de *apparatuur*, maar van de *programmatuur*. Met name in de theorievorming over *correctheid* van programma's, en hoe deze eigenschap precies te bewijzen, bleken begrippen en resultaten uit de logische semantiek en bewijstheorie van belang – waar men immers al had geleerd om systematisch om te gaan met geheel geformaliseerde talen. In hoofdstuk 15 wordt deze connectie verder behandeld, aan de hand van de zogenaamde Hoare–calculus. Overigens zijn verwante logische inzichten ook bruikbaar gebleken in de algebraïsche theorie van abstracte *datatypen* en bijbehorende *specificatietalen*. Dit thema laten we in dit boek onaangeroerd.
Een wat anders geaard modern contact vinden we in de theorie over *complexiteit* van programma's en berekenbaarheid. Complexiteit gaat over de hoeveelheid tijd of geheugen die nodig is om een in zo'n programma beschreven probleem op te lossen. Complexiteitstheorie gaat in hoofdlijnen terug op de wiskundige theorie van Turing-machines. Complexiteit wordt behandeld in hoofdstuk 17.

Gezien het formele, geheel precieze karakter van zowel logische formalismen als programmeertalen, zou een meer directe identificatie tussen beide wellicht voor de hand hebben gelegen. Niettemin is dit historisch niet gebeurd, met name omdat de gangbare programmeertalen een *sequentieel*, *imperatief* karakter hebben: ze beschrijven achtereenvolgende instructies. Dat karakter ontbreekt in de gangbare logische formalismen. Wel zijn gaandeweg zogenaamde *declaratieve* programmeertalen ontstaan, die wel heel expliciet van logische formalismen zijn afgeleid. Voorbeelden hiervan zijn PROLOG, dat afgeleid is van de predikaatlogica, en LISP, afgeleid van de lambda-calculus aangevuld met 'recursievergelijkingen'. Hoofdstuk 15 beschrijft meer in het algemeen het zogenaamde logisch programmeren van de eerste soort.

Ten slotte is er de laatste tijd een samenwerking op gang gekomen tussen de logica en de kunstmatige intelligentie, een tak van de informatica waar menselijke intelligente verrichtingen als waarneming, taalverwerving, redeneren of leren op de computer worden geanalyseerd en gesimuleerd. Gezien het gestelde in paragraaf 1.1, ligt dit geheel in het verlengde van het traditionele onderzoeksterrein van de logica. De hoofdstukken 18, 19 en 20 bieden een overzicht van enkele specifieke overeenkomsten, gekozen uit de gebieden van temporele datastructuren, multi-agentsystemen, en natuurlijke taalverwerking. We zullen hier toepassingen tegenkomen van predikaatlogica, modale logica en lambda–calculus.

Dit overzicht wil slechts aantonen dat er vele lijnen lopen tussen logica en informatica, die historisch ook goed te begrijpen zijn vanuit de ontwikkeling van deze beide vakken. De logica speelt hierbij een dubbele rol: enerzijds dienen logische systemen als voorbeeld voor *methoden* van programmeren en specificeren, anderzijds fungeert de logica ook als grondleggende *theorie* over belangrijke eigenschappen van deze activiteiten. Aldus heeft volgens velen de logica dezelfde rol verworven ten opzichte van de moderne informatica als bijvoorbeeld de wiskundige analyse reeds eeuwen speelt ten opzichte van de natuurkunde.

Behalve u vertrouwd te maken met de propositie– en predikaatlogica, is dit boek er dus op gericht u enige informatie te geven over logische gezichtspunten in de informatica en u daarmee ervaring te laten opdoen. Uiteraard zult u onderdelen uit dit boek weer bij diverse onderwerpen in de informatica zien terugkomen.

Blok II

Propositielogica

Propositielogica: syntaxis en semantiek

Hoofdstuk 2

Propositielogica: syntaxis en semantiek

2.1 INLEIDING

In het vorige hoofdstuk hebben we reeds twee voorbeelden gezien van een gevolgtrekking. Beide gevolgtrekkingen hebben dezelfde typerende globale vorm: er is een aantal *aannames*, gevolgd door een *conclusie:*

aanname	Als je hoofdpijn hebt, dan lust je deze chocola niet.
aanname	Je lust deze chocola.
conclusie	Je hebt geen hoofdpijn.

In dit geval is er sprake van twee aannames. In het algemeen kan er een willekeurig aantal aannames zijn. De volgende gevolgtrekking bestaat uit drie aannames en een conclusie:

aanname	Eva leest de krant en Jan doet de afwas.
aanname	Als Eva de krant leest, is de televisie uit.
aanname	Als Jan de afwas doet, neuriet hij.
conclusie	De televisie is uit en Jan neuriet.

Om gevolgtrekkingen nader te bestuderen, brengen we *structuur* aan in de aannames en de conclusie. Bepaalde (zins)delen van een gevolgtrekking kunnen zonder de geldigheid aan te tasten door andere vervangen worden. Als we in de eerste gevolgtrekking 'je hebt hoofdpijn' vervangen door 'je fiets is gestolen', dan ontstaat de volgende variant:

aanname	Als je fiets gestolen is, dan lust je deze chocola niet.
aanname	Je lust deze chocola.
conclusie	Je fiets is niet gestolen.

En ook 'je lust deze chocola' kan met behoud van geldigheid door iets anders vervangen worden. Dat wil zeggen: het vervangen van deze twee delen door willekeurige andere zinnen heeft geen invloed op de geldigheid van de gevolgtrekking. Om dit aan te geven, kunnen we 'je hebt hoofdpijn' en 'je lust chocola' vervangen door schematische letters: zeg h (om te herinneren aan de hoofdpijn) en c (chocola):

aanname	als h dan niet c
aanname	c
conclusie	niet h

Hiermee is de gevolgtrekking teruggebracht tot een overzichtelijker abstracte vorm.
We gaan nog een stapje verder. Naast de vervangbare delen bevat het voorgaande schema ook enkele onvervangbare logische 'sleuteltermen'. De vervanging daarvan zou wel degelijk tot verandering in geldigheid kunnen leiden; vervang bijvoorbeeld eens 'niet' door 'wel', of 'als ..., dan ...' door 'tenzij'! Ook voor deze onderdelen voeren we een korte symbolische notatie in, en wel voor 'als ..., dan ...' het symbool '$\rightarrow$' en voor 'niet' het symbool '$\neg$' (de haakjes zijn hulpsymbolen):

aanname	$(h \rightarrow \neg c)$
aanname	c
conclusie	$\neg h$

Deze gevolgtrekking noteren we nu kort als volgt:

$(h \rightarrow \neg c), c \;/\; \neg h$

Links van '/' staan de aannames, rechts ervan de conclusie.
In het algemeen, als een gevolgtrekking bestaat uit aannames $a_1, ..., a_n$ en conclusie c, dan noteren we deze gevolgtrekking als:

$a_1, ..., a_n \;\; / \; c$

Zolang de gevolgtrekking nog in gewone taal is geformuleerd, zet men overigens vaak aannames en conclusies onder elkaar, gescheiden door een horizontale streep. Deze notatie is dan overzichtelijker.

In het voorgaande hebben we Nederlandse zinnen door middel van bepaalde symbolen (h, c, $\rightarrow$, $\neg$), 'omgezet' in formele uitdrukkingen. Anders gezegd: de zinnen zijn 'vertaald' van een natuurlijke in een *formele* taal. Als we spreken over 'taal', dan spelen in het algemeen drie aspecten een belangrijke rol:

- Het *alfabet* geeft aan welke symbolen gebruikt mogen worden om uitdrukkingen (zinnen bijvoorbeeld) in de taal te maken.

- De *syntaxis* is de grammatica van een taal, dat wil zeggen, een geheel van regels dat aangeeft op welke manier uitdrukkingen van de taal gevormd mogen worden.
- De *semantiek* houdt zich bezig met de *betekenis* van uitdrukkingen: iemand kan wel weten hoe zinnen in het Japans gemaakt worden, maar als de betekenis niet bekend is, heb je er weinig aan.

Deze aspecten spelen bij natuurlijke talen een rol, maar ook bij programmeertalen en formele talen.

In dit hoofdstuk zullen we een formele taal introduceren, die de tot nu toe besproken gevolgtrekkingen kan verantwoorden. Dit houdt dus in dat elk van de drie genoemde aspecten van taal aan bod moet komen. Deze taal biedt enerzijds de mogelijkheid om uitdrukkingen uit de Nederlandse taal formeel te analyseren, maar kan anderzijds ook zelfstandig bestudeerd en ontwikkeld worden. Dit laatste houdt onder meer in dat we bepaalde *methoden* zullen introduceren om de geldigheid van gevolgtrekkingen te testen. Hierop komen we in de hoofdstukken 3 en 4 terug. Het aldus verkregen systeem, waarmee we in de komende hoofdstukken zullen leren werken, heet de *propositielogica*. In hoofdstuk 5 wordt ten slotte ingegaan op enkele belangrijke theoretische eigenschappen van de propositielogica als zodanig.

2.2 SYNTAXIS VAN DE PROPOSITIELOGICA

Het alfabet van de propositielogica kent drie soorten symbolen: *propositieletters*, *logische* symbolen en zogenaamde 'hulpsymbolen'.

Propositie

Een propositie is een bewering of uitspraak, uitgedrukt in een zin.

Voorbeeld 2.1

- In de natuurlijke taal: 'Lima is de hoofdstad van Peru', 'Jan ziet een krokodil', 'vannacht gaat de zomertijd in', 'alles is beter dan dit'.
- In de wiskunde: '$4 < 7$', '$\sqrt{25} = 5$', '$\pi \in Q$'.
- In de informatica: bijvoorbeeld een Boolese test, zoals 'X == 2'.

Propositieletter

In de formele taal gebruiken we *propositieletters* om proposities weer te geven die we niet nader wensen te analyseren. Notatie: kleine letters. Het is vaak handig om daarbij een herkenbare propositieletter te gebruiken: stel 'Jan huilt' voor door de letter h (van 'huilt'). Voor willekeurige, abstracte proposities gebruiken we bij voorkeur de letters p, q, r, ..., of soms ook p_1, p_2, p_3, ...

Logische symbolen

Naast de propositieletters zijn er vijf standaard logische symbolen. Met twee ervan hebben we reeds kennisgemaakt in paragraaf 2.1. Elk van deze symbolen heeft een uitspraak die meteen de betekenis van het symbool weergeeft. Hier volgen ze alle vijf :

$\neg$ niet
$\wedge$ en
$\vee$ of
$\rightarrow$ als ..., dan ...
$\leftrightarrow$ dan en slechts dan als

Deze symbolen worden vanwege hun functie ook wel *logische voegwoorden* of *connectieven* genoemd: ze kunnen proposities verbinden en op die manier samengestelde uitdrukkingen vormen.

Voorbeeld 2.2

Samengestelde zinnen uit de natuurlijke taal kunnen met logische symbolen worden ontleed:
- $(h \wedge l)$: 'Jan huilt en Jan loopt';
- $\neg h$: 'Jan huilt niet'.

De betekenis van de eerste vier symbolen zal geen problemen geven. Op de betekenis van $\leftrightarrow$ komen we later terug.

Hulpsymbolen

Naast de propositieletters en de logische symbolen zijn er nog hulpsymbolen. Dit zijn de haakjes (en), zoals in voorbeeld 2.2.

Meer formeel geven we nu de volgende definitie:

DEFINITIE 2.1

Alfabet
Een alfabet van de propositielogica bestaat uit:
a een verzameling propositieletters;
b logische symbolen: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$;
c hulpsymbolen:) en (.

Vervolgens kunnen we overgaan tot de syntaxis of grammatica.

Formule

Met symbolen uit het alfabet kunnen nu met behulp van vaste regels samengestelde uitdrukkingen in de propositielogica gemaakt worden. Die uitdrukkingen worden *formules* genoemd. In het algemeen gebruiken we kleine Griekse letters als zogenaamde formulevariabelen, om abstracte formules weer te geven.

DEFINITIE 2.2

Formule

De formules van de propositielogica worden als volgt gedefinieerd:
1 elke propositieletter is een formule;
2 als φ en ψ formules zijn, dan zijn $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ en $(\varphi \leftrightarrow \psi)$ ook formules;
3 niets anders is een formule.

De manier waarop we formules hebben gedefinieerd, heet *inductief*. Deze methode verdient enige aandacht op zich, omdat ze geregeld in dit boek zal terugkeren.

Inductieve definitie

Een *inductieve definitie* heeft drie kenmerkende onderdelen:
1 een *basisstap* waarin bepaalde dingen meteen tot objecten van de gewenste soort verklaard worden;
2 één of meer *opbouwstappen* die verdere constructieprincipes geven om objecten te maken;
3 een *afsluitende* stap die bepaalt dat alles wat niet in eindig veel stappen met behulp van 1 en 2 gevormd kan worden, geen toegestaan object is.

Atomaire formules

Propositieletters worden ook wel *atomaire formules* of *atomen* genoemd. Ze zijn namelijk niet verder te ontleden in kleinere formules.
De samengestelde formules hebben een vaste uitspraak en vormnaam:

vorm	*uitspraak*	*vormnaam*
$\neg\varphi$	niet φ	negatie
$(\varphi \wedge \psi)$	φ en ψ	conjunctie
$(\varphi \vee \psi)$	φ of ψ	disjunctie
$(\varphi \rightarrow \psi)$	als φ, dan ψ	implicatie
$(\varphi \leftrightarrow \psi)$	φ dan en slechts dan als ψ	equivalentie

Een negatie is dus een formule van de vorm $\neg\varphi$, enzovoorts.

Voorbeeld 2.3

Wel formule (we verwijzen steeds naar clausules uit definitie 2.2):
- p

(p is een propositieletter en dus volgens 1 een (atomaire) formule)
- $\neg\neg q$

(volgens 1 is q een formule; dan is volgens 2 $\neg q$ ook een formule en dan is, weer volgens 2, $\neg\neg q$ ook een formule)

- $\neg(p \rightarrow \neg q)$

(p en q zijn atomaire formules; dan is volgens 2 ook $\neg q$ een formule en volgens 2 is dan ook $(p \rightarrow \neg q)$ een formule; ten slotte is dan, weer volgens 2, ook $\neg(p \rightarrow \neg q)$ een formule)

- $(((p \wedge q) \rightarrow r) \rightarrow (\neg p \wedge q))$.

Geen formule:

- $(\neg p)$
- $q\neg$
- $(p \wedge q) \wedge r)$
- $(p \vee q \vee r)$

Deze 'symbolenrijtjes' kunnen namelijk niet met behulp van 1 en 2 gemaakt worden.

Formuleschema

Een vorm zoals $(\varphi \rightarrow \psi)$ is geen concrete formule maar een abstracte vorm van een formule. Dit noemen we een *formuleschema* (of soms kortweg toch ook 'formule'). Een concrete formule ontstaat als voor φ en ψ 'echte' formules ingevuld worden. Dit heet een *instantie* van het formuleschema.

Voorbeeld 2.4

Een aantal instanties van het formuleschema $(\varphi \rightarrow \psi)$ zijn:

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

(voor φ en ψ mag ook dezelfde formule ingevuld worden)

- $((p \wedge q) \rightarrow (q \vee (r \leftrightarrow \neg s)))$

(voor φ is $(p \wedge q)$ ingevuld, voor ψ de formule $(q \vee (r \leftrightarrow \neg s)))$

Als een formulevariabele meerdere keren voorkomt in een formuleschema, moet voor elk voorkomen hetzelfde ingevuld worden.

Voorbeeld 2.5

Twee instanties van $((\varphi \wedge \psi) \rightarrow \varphi)$:

- $((p \wedge q) \rightarrow p)$

(voor elk voorkomen van φ wordt p ingevuld)

- $(((p \rightarrow \neg q) \wedge (r \wedge \neg r)) \rightarrow (p \rightarrow \neg q))$

Achter deze voorbeelden steekt een algemeen syntactisch begrip:

Substitutie

Laat φ een formule zijn waarin (mogelijk) een propositieletter p voorkomt. Door elk voorkomen van p in φ te vervangen door een formule ψ, ontstaat een nieuwe formule. Het resultaat van dit *substitueren van ψ voor p in φ* wordt genoteerd als:

$[\psi/p]\varphi$

De eerder genoemde inductieve definitie van formules stelt ons in staat dit begrip simpel en overzichtelijk in te voeren via de opbouw van formules. We beschrijven weer het algemene patroon, omdat dit vaker zal voorkomen. Om een begrip uit te leggen voor alle objecten in een inductief gedefinieerde klasse objecten volstaat:

a het begrip uitleggen voor de basisobjecten;
b het begrip uitleggen voor complexe objecten, aangenomen dat het reeds is gedefinieerd voor de componenten.

De nu volgende omschrijving van substitutie, met 'inductie op de formule φ', illustreert het procédé:

DEFINITIE 2.3

Substitutie

a $[\psi/p]\varphi = \psi$ als $\varphi = p$
$[\psi/p]\varphi = \varphi$ als φ een propositieletter is en $\varphi \neq p$

b $[\psi/p]\neg\varphi = \neg[\psi/p]\varphi$
$[\psi/p](\varphi \vee \chi) = ([\psi/p]\varphi \vee [\psi/p]\chi)$
$[\psi/p](\varphi \wedge \chi) = ([\psi/p]\varphi \wedge [\psi/p]\chi)$
$[\psi/p](\varphi \rightarrow \chi) = ([\psi/p]\varphi \rightarrow [\psi/p]\chi)$
$[\psi/p](\varphi \leftrightarrow \chi) = ([\psi/p]\varphi \leftrightarrow [\psi/p]\chi)$

Voorbeeld 2.6

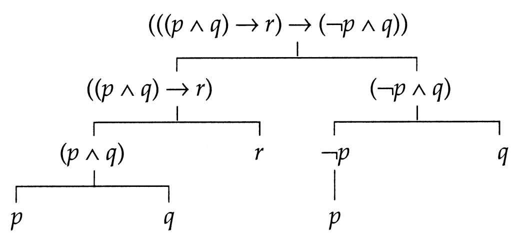
- $[(p \rightarrow q)/r](r \wedge s) = ((p \rightarrow q) \wedge s)$
- $[(p \vee \neg p)/q](q \rightarrow (s \rightarrow q)) = ((p \vee \neg p) \rightarrow (s \rightarrow (p \vee \neg p)))$

Constructieboom

Het opbouwen van een formule uit een aantal atomen met behulp van connectieven, kan worden weergegeven in een *constructieboom*. Zo'n constructieboom wordt van onder naar boven 'gelezen': we beginnen met de atomaire formules en bouwen daaruit de formule op.

Voorbeeld 2.7

De constructieboom voor de formule $(((p \wedge q) \rightarrow r) \rightarrow (\neg p \wedge q))$ ziet er als volgt uit:



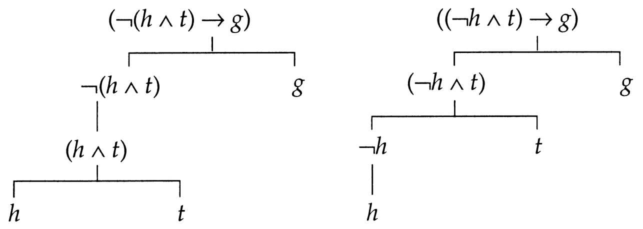
De haakjes leggen hier de constructie eenduidig vast. De formule in dit voorbeeld heeft dus alleen deze constructieboom. Ook in het algemeen valt te bewijzen dat formules precies één constructieboom hebben.

Vertalingen naar de propositielogica

Een zin in natuurlijke taal kan soms op meer dan één manier grammaticaal gelezen worden:

Voorbeeld 2.8

De zin 'Als de baby niet huilt en trappelt, dan is ze gelukkig' heeft drie verschillende lezingen:
a 'Als de baby niet zowel huilt als trappelt, dan is ze gelukkig';
b 'Als de baby niet huilt en (maar!) wel trappelt, dan is ze gelukkig';
c 'Als de baby niet huilt en niet trappelt, dan is ze gelukkig'.
Als we de propositieletters h, t en g invoeren voor respectievelijk 'de baby huilt', 'de baby trappelt' en 'de baby is gelukkig' dan zien de propositielogische vertalingen er als volgt uit:
a $(\neg(h \wedge t) \rightarrow g)$
b $((\neg h \wedge t) \rightarrow g)$
c $((\neg h \wedge \neg t) \rightarrow g)$
De eerste twee vertalingen verschillen alleen wat betreft de plaatsing van de haakjes. Hier zijn hun constructiebomen:



Haakjes zijn dus niet zomaar uit een formule weg te laten!

Bereik

De *haakjes* in een formule geven het *bereik* van een connectief aan. Het bereik van een voorkomen van een connectief is (zijn) de formule(s) die in de boom met dat connectief onmiddellijk verbonden wordt (worden). Het bereik van $\wedge$ in de eerste boom bestaat uit de formules h en t. Het bereik van $\wedge$ in de tweede boom bestaat uit de formules $\neg h$ en t. In de linkerboom is het bereik van $\neg$ de formule $(h \wedge t)$, in de rechterboom is het bereik van $\neg$ de (atomaire) formule h.

2.3 SEMANTIEK VAN DE PROPOSITIELOGICA

We weten nu hoe de syntaxis van de propositielogica eruitziet. Een formule was tot nu toe echter zonder betekenis: het is slechts een volgens bepaalde regels opgebouwd rijtje symbolen uit het alfabet van de taal. Wat is de betekenis van een formule?

Betekenis in natuurlijke taal

Laten we eens kijken naar de betekenis van een zin in natuurlijke taal. Beschouw de zin 'het regent'. Deze zin is in principe slechts een syntactisch rijtje symbolen. Stel nu dat het echt regent. We zeggen dan: de zin 'het regent' is *waar*. Als het niet regent, dan is 'het regent' *niet waar*. Met de begrippen *waar* en *niet waar* kan nu de betekenis van een zin vastgelegd worden. Als een zin waar is, dan wordt daarmee bedoeld dat wat in die zin uitgedrukt wordt, inderdaad het geval is. De *betekenis* van een zin wordt nu gegeven door te zeggen onder welke omstandigheden een zin waar is of niet waar.

Betekenis in propositielogica

Dit idee van betekenis gaat ook op voor formules. Net zoals een zin in natuurlijke taal waar of niet waar kan zijn, zeggen we ook dat een formule waar of niet waar kan zijn. De betekenis van een formule wordt dus gegeven door te zeggen wanneer die formule waar is of niet waar.

Waarheidswaarden

De begrippen 'waar' en 'niet waar' worden *waarheidswaarden* genoemd. In plaats van 'niet waar' gebruiken we ook vaak 'onwaar'. Verder wordt in plaats van 'waar' wel het cijfer 1 gebruikt en in plaats van 'niet waar' het cijfer 0.

Waarheidstabel

Voor atomen kunnen we waarheid of onwaarheid niet nader analyseren. We kunnen slechts aannemen dat ze waar of onwaar zijn. Maar hoe wordt nu de waarheidswaarde van een niet–atomaire formule bepaald? Beschouw de formule $(p \wedge q)$. Vergelijk deze formule met de zin 'Jan huilt en Piet lacht'. Deze zin is waar, als het waar is dat Jan huilt en het tegelijkertijd waar is dat Piet lacht, ofwel als 'Jan huilt' waar is en 'Piet lacht' ook. Als bijvoorbeeld Jan en Piet allebei lachen, dan is 'Jan huilt en Piet lacht' onwaar. De zin 'Jan huilt en Piet lacht' is dus waar als de twee zinnen die door 'en' verbonden worden, allebei waar zijn. Hetzelfde geldt ook voor $(p \wedge q)$. Deze formule is alleen waar als p en q beide waar zijn. Naast de mogelijkheid dat p en q beide waar zijn, zijn er nog drie andere mogelijkheden: (a) p is waar en q is onwaar, (b) p is onwaar en q is waar en (c) p en q zijn beide onwaar. In elk van deze drie gevallen is $(p \wedge q)$ niet waar.
Dit alles kan nu worden weergegeven in een tabel. Een dergelijke tabel heet een *waarheidstabel*. In zo'n tabel staan alle combinaties van de waarheidswaarden van p en q, met daarachter voor elk van deze gevallen de waarheidswaarde van $(p \wedge q)$ (voor de bondigheid gebruiken we in tabellen altijd 1 en 0):

p	q	$(p \wedge q)$
1	1	1
1	0	0
0	1	0
0	0	0

In de linkerkolom staan alle mogelijke combinaties van waarheidswaarden voor p en q. Elke rij in de tabel geeft dan aan wat de waarheidswaarde is van $(p \wedge q)$ bij gegeven waarheidswaarden van p en q. De derde rij bijvoorbeeld, zegt dat als p onwaar is en q waar, dan is $(p \wedge q)$ onwaar.

Waardering

Elke rij in een waarheidstabel correspondeert dus met een bepaald geval, een bepaalde *situatie*, waarin de waarheidswaarde van p en q dus vastligt (en daarmee ook die van $(p \wedge q)$). In de gegeven tabel zijn dat vier situaties, die alle mogelijkheden voor p en q uitputten. Zulke situaties noemen we *waarderingen*, aangegeven met V (van *valuation*). Waarderingen kunnen we beschouwen als functies van propositieletters naar waarheidswaarden.
De eerste situatie heeft bijvoorbeeld de volgende waardering:

$V(p) = 1$ en $V(q) = 1$

Dit wordt uitgesproken als: 'de waarde van p onder/bij de waardering V is 1'. Evenzo voor q. Volgens de tabel geldt dan ook:

$V((p \wedge q)) = 1$

Andere waarderingen kunnen kennelijk andere waarden opleveren. Soms is het handig om aan te geven dat een waardering bij een bepaalde situatie/rij hoort, hetgeen we kunnen aangeven door ze te nummeren: V_1, V_2, enzovoorts.

De waarheidstabellen van $\wedge$, $\neg$, $\vee$, $\rightarrow$ en $\leftrightarrow$

Ook voor willekeurige formules φ en ψ ziet de waarheidstabel van $(\varphi \wedge \psi)$er natuurlijk zo uit:

φ	ψ	$(\varphi \wedge \psi)$
1	1	1
1	0	0
0	1	0
0	0	0

Voor elk van de overige connectieven kan een dergelijke waarheidstabel gemaakt worden. Deze zien er als volgt uit:

φ	$\neg\varphi$
1	0
0	1

φ	ψ	$(\varphi \vee \psi)$
1	1	1
1	0	1
0	1	1
0	0	0

φ	ψ	$(\varphi \rightarrow \psi)$
1	1	1
1	0	0
0	1	1
0	0	1

ψ		$(\varphi \leftrightarrow \psi)$
1	1	1
1	0	0
0	1	0
0	0	1

Toelichting

- De tabel voor $\neg$ hoeft maar twee situaties te bevatten, omdat voor $\neg\varphi$ alleen de waarde van φ van belang is.
- Het is ook mogelijk om een andere tabel voor $\vee$ te geven. Dit ligt eraan hoe $\vee$ precies uitgelegd wordt. De mogelijkheid waarvoor we hier gekozen hebben, is dat $(\varphi \vee \psi)$ook waar is als φ en ψ beide waar zijn. Dit wordt de *inclusieve* uitleg van $\vee$ genoemd. De andere mogelijkheid is dat als φ en ψ beide waar zijn, dan $(\varphi \vee \psi)$niet waar is. Dit komt overeen met 'of ... of ...' in natuurlijke taal, waarmee bedoeld wordt 'of dit of dat, *maar niet allebei*'. Dit heet de *exclusieve* uitleg van $\vee$. In dit boek wordt, zoals algemeen gebruikelijk is, steeds gewerkt met de inclusieve uitleg.
- De implicatie $(\varphi \rightarrow \psi)$ is slechts in één situatie onwaar, namelijk als φ waar is en ψ onwaar. Als φ onwaar is, is de implicatie waar.

- De waarde van $(\varphi \leftrightarrow \psi)$ is alleen 1 als φ en ψ beide dezelfde waarde hebben (beide 1 of beide 0); met andere woorden, φ en ψ zijn 'gelijkwaardig' of 'equivalent'. Een equivalentie $(\varphi \leftrightarrow \psi)$ behelst intuïtief twee implicaties: $(\varphi \rightarrow \psi)$ en $(\psi \rightarrow \varphi)$.

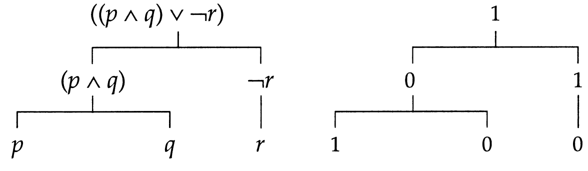
Compositionele semantiek

In de waarheidstabellen komt slechts één logisch connectief per keer voor. Hoe kan nu de waarheidswaarde van een willekeurige, eventueel meer complexe, formule bepaald worden? Beschouw bijvoorbeeld de formule $((p \wedge q) \vee \neg r)$. Om de waarheidswaarde hiervan te bepalen, moeten we eigenlijk de waarheidswaarden weten van $(p \wedge q)$ en van $\neg r$. Immers, als die bekend zijn, dan kan met de tabel voor $\vee$ de waarde van de hele formule bepaald worden. Maar wat is de waarde van $(p \wedge q)$ en $\neg r$? Om de waarde van $(p \wedge q)$ te bepalen, moeten we die van p en van q weten, want dan kan uit de tabel voor $\wedge$ de waarde van $(p \wedge q)$ afgelezen worden. Evenzo moeten we die van r kennen om de waarde van $\neg r$ te bepalen.
Waar dit alles op neerkomt, is het volgende. Het volstaat om een waardering te hebben voor de atomen die in een formule voorkomen; daarna levert een inductieve definitie stapsgewijs de uiteindelijke waarde, door per opbouwstap van de formule een bijpassende tabel toe te passen. Dit procédé van semantische evaluatie staat wel bekend als het principe van 'compositionaliteit van betekenis'. Dit is niet alleen een methodologisch beginsel voor de semantiek van formele talen, maar ook voor die van natuurlijke talen en programmeertalen. Dit zal blijken in latere hoofdstukken van dit boek.

Concreet laat zich de werking van het principe van compositionaliteit als volgt illustreren:

Voorbeeld 2.9

Zij V een waardering met $V(p) = 1$ en $V(q) = V(r) = 0$. Beschouw de syntactische constructieboom van de formule $((p \wedge q) \vee \neg r)$. We kunnen nu de waarderingen voor zijn subformules van onder naar boven berekenen:



Waarheidstabel van een formule

Door nu alle mogelijke waarderingen voor de atomen van een formule in een tabel onder elkaar te zetten, kan met behulp van de tabellen voor de connectieven de waarheidswaarde van een formule in alle situaties bepaald worden.

Voorbeeld 2.10

Voor $((p \wedge q) \vee \neg r)$ ziet zo'n tabel er als volgt uit:

	p	q	r	$(p \wedge q)$	$\neg r$	$((p \wedge q) \vee \neg r)$
V_1	1	1	1	1	0	1
V_2	1	1	0	1	1	1
V_3	1	0	1	0	0	0
V_4	1	0	0	0	1	1
V_5	0	1	1	0	0	0
V_6	0	1	0	0	1	1
V_7	0	0	1	0	0	0
V_8	0	0	0	0	1	1

In de linkerkolom staan de verschillende waarderingen onder elkaar; vervolgens worden per waardering de waarden van $(p \wedge q)$ en $\neg r$ bepaald en daarmee weer die van $((p \wedge q) \vee \neg r)$.

Ook dit soort tabellen heten waarheidstabellen. Als er n verschillende atomen in een formule φ voorkomen, dan zijn er 2^n verschillende waarderingen. Voor ieder atoom zijn er immers twee mogelijkheden, waar of onwaar. We zeggen ook wel: φ heeft 2^n *relevante* waarderingen. Als er veel atomen zijn, is het verstandig om deze waarderingen op een systematische manier op te sommen, zoals in voorgaande tabel. Waarheidstabellen kunnen ook iets compacter worden genoteerd. In de linkerkolom staan weer de verschillende waarderingen, maar in de rechterkolom staat alleen de hele formule waarvan de waarheidswaarde bepaald moet worden. De berekende waarden van subformules die in deze formule voorkomen, worden direct onder het connectief gezet dat die subformule gevormd heeft. Toegepast op voorbeeld 2.10 levert dit:

Voorbeeld 2.11

	p	q	r	$((p \wedge q) \vee \neg r)$		
V_1	1	1	1	1	1	0
V_2	1	1	0	1	1	1
V_3	1	0	1	0	0	0
V_4	1	0	0	0	1	1
V_5	0	1	1	0	0	0
V_6	0	1	0	0	1	1
V_7	0	0	1	0	0	0
V_8	0	0	0	0	1	1

Ook voor een formuleschema kan zo'n waarheidstabel gemaakt worden. Zo kunnen in voorbeeld 2.11 p, q en r zonder bezwaar vervangen worden door respectievelijk φ, ψ en χ.

2.4 MODELLEN

De betekenis van een formule wordt gegeven door het verloop van haar waarheidswaarden over de relevante waarderingen. Strikt genomen hoeven we hierbij alleen te kijken naar de waarden van de propositieletters die in die formule voorkomen. Willen we echter ineens over betekenis van de hele taal spreken, dan dient het begrip 'waardering' ruimer opgevat te worden, en wel als volgt:

DEFINITIE 2.4

Waardering
Een waardering is een functie van alle propositieletters naar waarheidswaarden.

Een waardering geeft dus aan *elke* propositieletter de waarde 0 of 1. Zo kan bij een gegeven waardering de waarheidswaarde van een willekeurige formule φ bepaald worden. Immers, een waardering geeft aan elke propositieletter een waarde, dus ook de propositieletters (atomen) in φ. We blijven vaak de eerdere definitie met beperkt domein gebruiken.

Model

We kunnen de semantiek ook anders verbeelden. Bij een gegeven aantal propositieletters hoort een aantal relevante waarderingen. Die vormen samen de ruimte van alle *mogelijke* situaties. De *feitelijke* situatie is één van deze mogelijke situaties. Deelverzamelingen van de situatieruimte corresponderen met 'informatietoestanden'. De hele verzameling geeft

geen informatie over wat de feitelijke situatie is. Een deelverzameling kan gezien worden als een benadering van de feitelijke situatie en geeft daarom wel informatie. Een deelverzameling met één element legt een situatie vast en geeft dus volledige informatie.
Deze informatieve inhoud van een formule is in de logica formeel gemaakt. Men spreekt van de verzameling *modellen* van een formule.

DEFINITIE 2.5

Model
Een waardering V heet een *model* van een formule φ als geldt: $V(\varphi) = 1$.

Voorbeeld 2.12

De formule $((p \wedge q) \vee \neg r)$ uit voorbeeld 2.11 heeft acht relevante waarderingen, waaronder vijf modellen: V_1, V_2, V_4, V_6 en V_8.

De verzameling modellen van een formule φ noteren we als volgt:

$$MOD(\varphi) = \{V \mid V(\varphi) = 1\}$$

We kunnen ook spreken over de modellen van een *verzameling* formules:

DEFINITIE 2.6

Model van een formuleverzameling
Een waardering V heet een model van een formuleverzameling Σ als zij model is van elke $\varphi \in \Sigma$.

De verzameling modellen van een formuleverzameling Σ noteren we als $MOD(\Sigma)$. Als we Σ uitbreiden met een formule φ, dan geldt $MOD(\Sigma \cup \{\varphi\}) \subseteq MOD(\Sigma)$. Immers, een model van $\Sigma \cup \{\varphi\}$ is een model V van Σ waarvoor tevens geldt $V(\varphi) = 1$.
Voor een formule of een formuleverzameling associëren we veel modellen dus met weinig informatieve inhoud, weinig modellen met veel informatieve inhoud, en één model met volledige informatie.

Modeleliminatie

Een beeld van informatieverwerking dat de laatste jaren sterk opgang maakt binnen de informatica en de taalkunde, is het volgende proces van 'eliminatie van semantische alternatieven'.
Stel Σ is een formuleverzameling, waarbij $MOD(\Sigma)$ onze huidige kennistoestand representeert. Stel dat we nu een nieuwe bewering φ erbij leren. De extra informatie die dit oplevert, zit dan in de overgang van de verzameling $MOD(\Sigma)$ naar haar deelverzameling $MOD(\Sigma \cup \{\varphi\})$. Voortzetting van dit proces kan leiden tot een uniek model, vastgelegd in een unieke waardering, met andere woorden volledige kennis over de situatie. Dit lichten we toe aan de hand van het volgende voorbeeld:

Voorbeeld 2.13

Beschouw de volgende zinnen:
a 'Jan komt als Marie of Anne komt'
b 'Anne komt als Marie niet komt'
c 'Jan komt niet als Anne komt'

Verschaft dit ons nu voldoende informatie voor het beantwoorden van de vraag 'wie komt er wel en wie niet'?
Met behulp van het gestelde over modeleliminatie kan deze vraag beantwoord worden. We beginnen met een voor de hand liggende symbolische weergave:

a $((m \vee a) \rightarrow j)$ (noem dit φ)
b $(\neg m \rightarrow a)$ (ψ)
c $(a \rightarrow \neg j)$ (χ)

De gestelde vraag komt nu neer op: heeft $\{\varphi, \psi, \chi\}$ een uniek model? Om hier achter te komen, bepalen we eerst alle modellen van φ, vervolgens kijken we welke van deze modellen tevens modellen zijn van ψ, en ten slotte kijken we welke van de resterende modellen nog modellen zijn van χ. Dit levert een reductie van 8 situaties aan het begin, via 5 tot 3 tot inderdaad 1, als in de volgende tabel:

waarderingen			*model van*		
m	a	j	$\{\varphi\}$	$\{\varphi, \psi\}$	$\{\varphi, \psi, \chi\}$
1	1	1	ja	ja	nee
1	1	0	nee	–	–
1	0	1	ja	ja	ja
1	0	0	nee	–	–
0	1	1	ja	ja	nee
0	1	0	nee	–	–
0	0	1	ja	nee	–
0	0	0	ja	nee	–

Dit 'zeven' van modellen leidt ertoe dat één waardering overblijft als model voor $\{\varphi, \psi, \chi\}$, namelijk $V(m) = V(j) = 1$ en $V(a) = 0$. Ofwel: Marie en Jan komen, Anne niet.

Tautologie

Doorgaans zal voor een formule φ niet elke waardering model zijn. Wanneer dit wel het geval is, heet φ een tautologie:

DEFINITIE 2.7

Tautologie
Een formule φ heet een *tautologie* als elke waardering een model is van φ, ofwel als voor elke waardering V geldt: $V(\varphi) = 1$.

Voorbeeld 2.14

Elke formule van de vorm $(\varphi \vee \neg\varphi)$ is een tautologie:

φ	$\neg\varphi$	$(\varphi \vee \neg\varphi)$
1	0	1
0	1	1

Andere voorbeelden van tautologieën zijn $((\neg(\varphi \wedge \psi) \wedge \varphi) \to \neg\psi)$ en $\neg(\varphi \wedge \neg\varphi)$. De informatieve inhoud van een tautologie is nihil, omdat ieder model een model is van een tautologie. Toch kunnen tautologieën, anders dan men nu wellicht zou denken, nuttige principes uitdrukken:

DEFINITIE 2.8

Logisch equivalent
Twee formules φ en ψ heten *logisch equivalent* als de formule $(\varphi \leftrightarrow \psi)$een tautologie is.

Voorbeeld 2.15

De formules $\neg\neg p$ en p zijn logisch equivalent:

p	$\neg\neg p$	$(p \leftrightarrow \neg\neg p)$
1	1	1
0	0	1

Voorbeeld 2.16

Andere voorbeelden van veel gebruikte logische equivalenties:
- $(\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q))$
- $(\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q))$
- $((p \wedge (q \vee r)) \leftrightarrow ((p \wedge q) \vee (p \wedge r)))$
- $((p \vee (q \wedge r)) \leftrightarrow ((p \vee q) \wedge (p \vee r)))$

De eerste twee equivalenties worden wel de 'wetten van De Morgan' genoemd, de laatste twee zijn de zogenaamde 'principes van distributiviteit'.

Associativiteit en haakjesbesparing

Logische equivalenties rechtvaardigen soms ook verruimingen van notatie. Op grond van de tautologie $(((\varphi \wedge \psi) \wedge \chi) \leftrightarrow (\varphi \wedge (\psi \wedge \chi)))$ staan we voortaan toe om in willekeurig lange conjuncties alle haakjes behalve de buitenste weg te laten. Dit noemen we de *associativiteit* van het connectief $\wedge$. De interpretatie van zo'n formule blijkt immers onafhankelijk van de plaatsing van haakjes erbinnen. Ook het connectief $\vee$ is associatief.
Buitenste haakjes laten we in het vervolg eveneens vaak weg bij formules, tenzij daardoor verwarring kan ontstaan over de constructieboom.

Voorbeeld 2.17

De formule $(((p \wedge q) \wedge r) \wedge s)$ mag ook geschreven worden als $p \wedge q \wedge r \wedge s$.

Opsomming van modellen

Beschouw nogmaals de formule $((p \wedge q) \vee \neg r)$ uit voorbeeld 2.9. Deze formule had acht relevante situaties, waaronder vijf modellen (zie de waarheidstabel). In V_1 geldt bijvoorbeeld: p is waar en q is waar en r is waar. We kunnen deze situatie kennelijk karakteriseren door de formule $p \wedge q \wedge r$. Evenzo geldt dat V_2 wordt gekarakteriseerd door $p \wedge q \wedge \neg r$, enzovoorts. Maar dan kunnen we $((p \wedge q) \vee \neg r)$ ook logisch equivalent omschrijven door de volgende disjunctieve opsomming van zijn vijf modellen:

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge \neg r)$$

Zo'n opsomming van modellen kan voor iedere formule worden gemaakt. We kunnen hierbij twee kanttekeningen van algemene strekking maken:

- in de opsomming komen slechts de connectieven $\wedge$, $\vee$ en $\neg$ voor;
- de opsomming is een disjunctie van conjuncties van atomen of negaties van atomen.

Functionele volledigheid

De eerste opmerking leidt ons tot de volgende definitie:

DEFINITIE 2.9

Functioneel volledig
Laat C een verzameling connectieven zijn (bijvoorbeeld $\{\neg, \vee\}$ of $\{\vee, \wedge, \leftrightarrow\}$). C heet *functioneel volledig* als elke formule φ logisch equivalent is met een formule ψ die slechts connectieven uit C bevat.

Uit de voorgaande beschouwing blijkt dus dat $\{\neg, \wedge, \vee\}$ functioneel volledig is. Zelfs $\{\neg, \vee\}$ is functioneel volledig, op grond van de tautologie $(\varphi \wedge \psi) \leftrightarrow \neg(\neg\varphi \vee \neg\psi)$. Hiermee kunnen we immers in iedere

formule alle voorkomens van het connectief $\wedge$ vervangen door een combinatie van $\neg$ en $\vee$. Niet volledig is daarentegen $\{\neg, \leftrightarrow\}$ of $\{\wedge, \vee\}$. Formele bewijzen van functionele volledigheid worden in hoofdstuk 5 gegeven.
Er zijn enkele connectieven die op zichzelf functioneel volledig zijn.

Voorbeeld 2.18

Het NOR–connectief
Een voorbeeld hiervan is het tweeplaatsige zogeheten NOR–connectief ('noch ... noch ...'), dat de volgende waarheidstabel heeft:

φ	ψ	φ NOR ψ
1	1	0
1	0	0
0	1	0
0	0	1

Waarheidsfunctie

In feite kunnen we onze analyse nog iets generaliseren. Blijkbaar kunnen we andere dan standaardconnectieven invoeren, zoals de exclusieve disjunctie of de NOR. Omgekeerd kunnen we met elk van de zestien mogelijke waarheidstabellen een tweeplaatsig connectief associëren. Nog algemener hoort bij iedere invulling van een kolom waarheidswaarden voor een complete verzameling waarderingen het volgende begrip:

DEFINITIE 2.10

Waarheidsfunctie
Een functie van $\{0, 1\}^k$ naar $\{0, 1\}$ heet een k–plaatsige waarheidsfunctie.

Met het eerdergenoemde opsommingsargument blijkt iedere waarheidsfunctie definieerbaar door een formule waarin standaard-connectieven worden gebruikt.

Disjunctieve normaalvorm

We zagen dat de opsomming van modellen van een formule een speciale syntactische vorm heeft, namelijk een disjunctie van conjuncties, bestaande uit atomen of negaties van atomen. We spreken dan van een *disjunctieve normaalvorm*:

$$(\varphi_1 \wedge \ldots \wedge \varphi_{n_1}) \vee \ldots \vee (\chi_1 \wedge \ldots \wedge \chi_{n_k})$$

waarbij $\varphi_1, \ldots, \chi_{n_k}$ atomen of negaties van atomen zijn.

Het voorgaande bewijst dus ook het volgende resultaat:

Als φ een formule is, dan bestaat er een formule φ^* in disjunctieve normaalvorm, zodat φ en φ^* logisch equivalent zijn.

Normaalvormen zijn speciale syntactische vormen waarmee meer algemene vormen equivalent kunnen zijn. Door hun uniformiteit zijn ze nuttig voor diverse berekeningsdoeleinden. We zullen ze ook in latere hoofdstukken tegenkomen.

2.5 TOT BESLUIT

Hiermee is de taal van de propositielogica voldoende ingevoerd. In het voorgaande zijn er al enkele meer theoretische vragen gerezen die om beantwoording vragen. Hoofdstuk 5 biedt daarover nader uitsluitsel en geeft methoden om resultaten *over* de propositielogica te bewijzen.
Wij besluiten hier met een andere opmerking. Abstract werken is vaak profijtelijk. Zoals zoveel wiskundige formalismen blijkt de propositielogica diverse toepassingen te bezitten buiten haar oorspronkelijke motivering. De eerder genoemde principes van distributiviteit bijvoorbeeld kunnen op minstens drie manieren geïnterpreteerd worden:

Voorbeeld 2.19

Drie visies op distributie
De tautologie $(p \wedge (q \vee r)) \leftrightarrow ((p \wedge q) \vee (p \wedge r))$ drukt achtereenvolgens uit:

- een logische equivalentie tussen *beweringen* (de oorspronkelijke motivering)

maar ook:

- een wiskundige gelijkheid tussen *verzamelingen*: p doorsneden met de vereniging van q en r is dezelfde verzameling als de vereniging van de doorsnede van p en q met die van p en r

en net zo goed nog eens:

- een natuurkundig principe van '*schakelalgebra*': een serieschakeling van een schakelaar p met een parallelschakeling van q en r heeft precies hetzelfde elektrische effect als de parallelschakeling van de serie p, q met de serie p, r.

Op details van deze interpretaties gaan we hier niet verder in.

2.6 OPGAVEN

2.1 Geef de volgende zinnen weer in propositionele notatie:
i 'Als de bus niet komt, komen de tram en de trein'.
ii 'Als de tram komt als de trein niet komt, dan komen de trein en de bus niet allebei'.

2.2 Geef alle formules die met haakjes invoegen zijn te maken uit $p \wedge \neg q \rightarrow r$, met de bijbehorende constructiebomen.

2.3 Welke getallen x voldoen aan de volgende condities:
i $\neg(p \wedge q) \wedge r$
ii $(\neg p \wedge q) \wedge r$
iii $\neg(p \wedge (q \wedge r))$
Hierbij geldt p = '$x \leq 1$', q = '$x \leq 3$' en r = '$x \geq 2$'.

2.4 Maak een waarheidstabel voor elk van de volgende connectieven:
i *noch* φ *noch* ψ
ii φ *alleen als* ψ
iii φ *tenzij* ψ

2.5 Geef een disjunctieve normaalvorm voor $((p \wedge q) \rightarrow r) \rightarrow (\neg p \wedge q)$.

2.6 Bepaal alle niet logisch equivalente waarheidsfuncties die te definiëren zijn met behulp van twee propositieletters p, q en alleen het connectief $\rightarrow$. (Eén hiervan is de disjunctie $\vee$.)

Steropgaven

De opgaven gemerkt met * zijn moeilijker dan de opgaven zonder *.

* 2.7 In de informatica definieert men programmeertalen vaak via de zogenaamde *Backus–Naur form* (BNF). Voor propositionele formules is de BNF–notatie als volgt (waarbij F staat voor 'Formule'):

$$F := ATOM \mid \neg F \mid F \wedge F \mid F \vee F \mid F \rightarrow F \mid F \leftrightarrow F$$

Laat zien dat dit dezelfde verzameling oplevert als definitie 2.2.

* 2.8 Laat zien hoe het bekende spel 'Master Mind' is op te vatten als een eliminatieproces van modellen zoals in de tekst beschreven. Wat kiest u als propositionele atomen, hoe geeft u de antwoorden weer, wat is het maximaal aantal nodige vragen?

Inhoud hoofdstuk 3

Propositielogica: geldig gevolg

Hoofdstuk 3

Propositielogica: geldig gevolg

3.1 GELDIG GEVOLG

Geldig

In hoofdstuk 1 werd de notie 'geldigheid' van een gevolgtrekking informeel ingevoerd. Met behulp van de semantiek uit hoofdstuk 2 kunnen we dit nu preciseren.

DEFINITIE 3.1

Geldig gevolg
Een formule ψ heet een *geldig gevolg* van een verzameling formules Σ als elk model van Σ ook model is van ψ.

Als ψ een geldig gevolg is van Σ, dan wordt dit genoteerd als:

$$\Sigma \vDash \psi$$

Anders gezegd: 'de gevolgtrekking Σ / ψ is geldig'. Als $\Sigma = \{\varphi_1, ..., \varphi_n\}$, dan wordt in plaats van $\Sigma \vDash \psi$ ook wel $\varphi_1, ..., \varphi_n \vDash \psi$ geschreven. Of anders: '$\varphi_1, ..., \varphi_n / \psi$ is geldig'.

Voorbeeld 3.1

- $p, p \rightarrow q \vDash q$
- $p, p \rightarrow q, q \rightarrow r \vDash r$

Een speciaal geval van definitie 3.1 ontstaat als $\Sigma = \varnothing$. Dit geeft:

$$\vDash \psi$$

Als $\Sigma = \varnothing$, dan is elke waardering model van Σ. Dus $\vDash \psi$ betekent dat ψ waar is onder elke waardering, ofwel: ψ is een tautologie (zie definitie 2.7).
Als ψ niet een geldig gevolg is van Σ, dan noteren we dit als:

$$\Sigma \nvDash \psi$$

Dit betekent dat er een model van Σ bestaat dat geen model van ψ is. Zo'n model heet een *tegenvoorbeeld* van de gevolgtrekking $\Sigma \,/\, \psi$.

Voorbeeld 3.2

De gevolgtrekking $q, p \rightarrow q \;/\; p$ is niet geldig. Als bijvoorbeeld $V(q) = 1$ en $V(p) = 0$, dan $V(p \rightarrow q) = 1$ (volgens de waarheidstabel van $\rightarrow$). Deze waardering V maakt dus q en $p \rightarrow q$ waar, maar p niet.

Geldigheid en waarheid

Het is belangrijk om de begrippen 'geldigheid' en 'waarheid' niet met elkaar te verwarren. Geldige gevolgtrekkingen kunnen in sommige situaties onware aannames of conclusies hebben. Zo is de gevolgtrekking $p \vee q, \neg q \wedge r \;/\; p \wedge r$ geldig. Tegelijkertijd is er een waardering V die alle drie de betrokken formules onwaar maakt, namelijk $V(p) = V(q) = V(r) = 0$.

Hoe kunnen we er nu achter komen of een gegeven gevolgtrekking geldig is? Eén manier is het opstellen van een waarheidstabel voor de in de gevolgtrekking voorkomende formules.

Voorbeeld 3.3

Is de gevolgtrekking $\neg q, p \rightarrow q \;/\; \neg p$ geldig? We maken een waarheidstabel waarin de waarden van de formules $\neg q$, $p \rightarrow q$ en $\neg p$ berekend worden:

	p	q	$\neg q$	$p \rightarrow q$	$\neg p$
V_1	1	1	0	1	0
V_2	1	0	1	0	0
V_3	0	1	0	1	1
V_4	0	0	1	1	1

Uit deze tabel blijkt dat $\{\neg q, p \rightarrow q\}$ slechts één model heeft, namelijk V_4, en dat V_4 ook een model is van $\neg p$. Dus de gevolgtrekking is geldig: $\neg q, p \rightarrow q \models \neg p$.

Een nadeel van deze methode is dat het aantal waarderingen exponentieel stijgt met het aantal atomen dat voorkomt in de gevolgtrekking. Het is vaak efficiënter om gericht naar een tegenvoorbeeld te zoeken. Dit doen we met zogenaamde 'semantische tableaus'.

3.2 SEMANTISCHE TABLEAUS

Sequent

We beginnen met een nieuw begrip. Een *sequent* is een rijtje van de volgende vorm, waarbij $\varphi_1, ..., \varphi_n$ en $\psi_1, ..., \psi_m$ twee rijtjes formules zijn, die worden gescheiden door een nieuw symbool $\circ$:

$$\varphi_1, ..., \varphi_n \circ \psi_1, ..., \psi_m$$

Een waardering V heet een *tegenvoorbeeld* van een sequent $\varphi_1, ..., \varphi_n \circ \psi_1, ..., \psi_m$ indien $V(\varphi_1) = ... = V(\varphi_n) = 1$ en $V(\psi_1) = ... = V(\psi_m) = 0$. Merk op dat een sequent waarin een formule φ aan beide kanten van $\circ$ voorkomt, geen tegenvoorbeeld kan hebben.

Semantisch tableau

Een *semantisch tableau* is een schema waarin op systematische wijze het mogelijk bestaan van tegenvoorbeelden van een gegeven sequent teruggebracht (*gereduceerd*) wordt tot dat van één of meer overzichtelijker sequenten. Uiteindelijk voert dit proces ons tot de eenvoudigste sequenten, waarvoor onze vraag meteen valt te beslissen.
Bepalen of een gevolgtrekking $\varphi_1, ..., \varphi_n \,/\, \psi$ een tegenvoorbeeld heeft, doen we dan door een semantisch tableau te maken voor de sequent $\varphi_1, ..., \varphi_n \circ \psi$. Enkele eenvoudige gevallen maken de methode duidelijk:

Voorbeeld 3.4

De geldigheid van de gevolgtrekking $\neg q, p \rightarrow q \,/\, \neg p$ uit voorbeeld 3.3 zien we als volgt in. Stel dat een zekere waardering V de proposities $\neg q$ en $p \rightarrow q$ waar maakt, en $\neg p$ onwaar:

$$\neg q, p \rightarrow q \circ \neg p$$

Deze bewering valt te vereenvoudigen. Immers, $\neg q$ waar maken, is hetzelfde – volgens de waarheidstabel voor negatie – als q onwaar maken. Dus mogen we ook noteren:

$$p \rightarrow q \circ \neg p, q$$

Maar evenzo is $\neg p$ onwaar maken hetzelfde als p waar maken:

$$p \rightarrow q, p \circ q$$

Aldus is de complexiteit van de vraagstelling al met twee connectieven gereduceerd. We gaan over tot het resterende connectief. Thans doet zich een *keuze* voor. Blijkens de waarheidstabel voor implicatie zijn er twee verschillende manieren om $p \rightarrow q$ waar te maken, te weten: q waar maken of p onwaar maken. Deze schrijven we beide op:

$q, p \circ q \qquad p \circ q, p$

Maar nu kunnen we meteen constateren dat beide mogelijkheden falen, aangezien er telkens een formule zowel links als rechts voorkomt. Dus elke mogelijkheid om een tegenvoorbeeld te construeren heeft gefaald: de oorspronkelijke gevolgtrekking moet wel geldig zijn.

Wanneer we dezelfde gedachtengang zouden volgen bij de variant $\neg p, p \rightarrow q$ / $\neg q$, dan arriveren we juist bij de sequent

$p \rightarrow q, q \circ p$

die aanleiding geeft tot twee atomaire gevallen

$q, q \circ p \qquad q \circ p, p$

die beide een tegenvoorbeeld opleveren, en wel hetzelfde: $V(p) = 0$, $V(q) = 1$. De gevolgtrekking $\neg p, p \rightarrow q$ / $\neg q$ is dus ongeldig.

Nog andere reductieregels komen we tegen wanneer we eenzelfde analyse uitvoeren op geldige gevolgtrekkingen als $\neg(p \wedge q), p \vee r$ / $\neg q \vee r$. Het waar maken van een disjunctie kan op twee manieren: het linkerlid waar maken of het rechterlid waar maken. Het onwaar maken van een disjunctie kan slechts op één manier: zowel linker– als rechterlid onwaar maken.

We beschrijven nu de methode preciezer. Hierin gebruiken we de volgende notatie voor een rijtje formules: Φ voor het rijtje $\varphi_1, ..., \varphi_n$ en Ψ voor het rijtje $\psi_1, ..., \psi_m$. Deze notatie voor een *rijtje* en het vergelijkbare gebruik van Griekse hoofdletters voor een *verzameling* formules zullen we soms door elkaar heen gebruiken in volgende hoofdstukken. Uit de context blijkt dan of er verzamelingen dan wel rijtjes bedoeld worden. Eenzelfde verhaal geldt natuurlijk voor notaties als '$\Sigma \cup \{\varphi\}$', '$\Sigma + \varphi$' en 'Σ, φ': alleen in het laatste geval staat Σ voor een rijtje.

Reductieregels

Een sequent wordt gereduceerd met behulp van *reductieregels*. Een reductieregel heeft een van de volgende twee vormen:

$$\begin{array}{c} \Phi \circ \Psi \\ | \\ \Phi' \circ \Psi' \end{array} \qquad\qquad \begin{array}{cc} \multicolumn{2}{c}{\Phi \circ \Psi} \\ \multicolumn{2}{c}{\overbrace{\hspace{8em}}} \\ \Phi' \circ \Psi' & \Phi'' \circ \Psi'' \end{array}$$

De linkervorm wordt als volgt gelezen:

$\Phi \circ \Psi$ heeft een tegenvoorbeeld desda $\Phi' \circ \Psi'$ heeft een tegenvoorbeeld.

Notatie

'Desda' staat voor 'dan en slechts dan als', de tekstuele vorm van de dubbele implicatie $\Leftrightarrow$ die u uit de wiskunde kent. Op de relatie tussen $\Leftrightarrow$ en het propositielogische symbool $\leftrightarrow$ komen we in hoofdstuk 5 terug.

De 'splitsende' rechtervorm betekent:

$\Phi \circ \Psi$ heeft een tegenvoorbeeld desda $\Phi' \circ \Psi'$ heeft een tegenvoorbeeld *of* $\Phi'' \circ \Psi''$ heeft een tegenvoorbeeld.

Bij elk connectief hoort een linker en een rechter reductieregel: in de linker reductieregel komt het connectief aan de linkerkant van het teken $\circ$ voor, in de rechter reductieregel aan de rechterkant. De regels voor het connectief $\neg$ zijn kennelijk als volgt:

$$\neg_L: \quad \begin{array}{c} \Phi, \neg\alpha \circ \Psi \\ | \\ \Phi \circ \alpha, \Psi \end{array} \qquad\qquad \neg_R: \quad \begin{array}{c} \Phi \circ \neg\alpha, \Psi \\ | \\ \Phi, \alpha \circ \Psi \end{array}$$

Het is eenvoudig in te zien dat dit correct is. Stel bijvoorbeeld dat V een waardering is zodat $V(\Phi) = 1$, $V(\neg\alpha) = 1$ en $V(\Psi) = 0$ (met $V(\Phi) = 1$ wordt bedoeld $V(\varphi) = 1$, voor elke φ die in Φ voorkomt). Dan is dit equivalent met: $V(\Phi) = 1$, $V(\alpha) = 0$ en $V(\Psi) = 0$. Op eenzelfde manier is de juistheid van $\neg_R$ in te zien. Een soortgelijke motivering als voor $\neg$ valt te geven bij de reductieregels voor de overige connectieven, die we nu tabelleren:

$$\wedge_L:\quad \dfrac{\Phi, \alpha \wedge \beta \circ \Psi}{\Phi, \alpha, \beta \circ \Psi} \qquad\qquad \wedge_R:\quad \dfrac{\Phi \circ \alpha \wedge \beta, \Psi}{\Phi \circ \alpha, \Psi \qquad \Phi \circ \beta, \Psi}$$

$$\vee_L:\quad \dfrac{\Phi, \alpha \vee \beta \circ \Psi}{\Phi, \alpha \circ \Psi \qquad \Phi, \beta \circ \Psi} \qquad\qquad \vee_R:\quad \dfrac{\Phi \circ \alpha \vee \beta, \Psi}{\Phi \circ \alpha, \beta, \Psi}$$

$$\rightarrow_L:\quad \dfrac{\Phi, \alpha \rightarrow \beta \circ \Psi}{\Phi, \beta \circ \Psi \qquad \Phi \circ \alpha, \Psi} \qquad\qquad \rightarrow_R:\quad \dfrac{\Phi \circ \alpha \rightarrow \beta, \Psi}{\Phi, \alpha \circ \beta, \Psi}$$

$$\leftrightarrow_L:\quad \dfrac{\Phi, \alpha \leftrightarrow \beta \circ \Psi}{\Phi, \alpha, \beta \circ \Psi \qquad \Phi \circ \alpha, \beta, \Psi} \qquad\qquad \leftrightarrow_R:\quad \dfrac{\Phi \circ \alpha \leftrightarrow \beta, \Psi}{\Phi, \alpha \circ \beta, \Psi \qquad \Phi, \beta \circ \alpha, \Psi}$$

Met behulp van een reductieregel kan een gegeven sequent S gereduceerd worden tot één of twee andere sequenten. Deze kunnen op hun beurt weer verder gereduceerd worden, enzovoorts. Aldus ontstaat een boomvorm, met S als wortel en sequenten $S_1, ..., S_n$ als bladeren. Als zo'n boom niet verder gereduceerd kan worden, spreken we van een *semantisch tableau*. Merk op dat een tableau altijd eindig is: de vertakkingsgraad van de boom is ten hoogste 2, terwijl de taklengte wordt begrensd door het totaal aantal connectieven dat in de topsequent voorkomt.

Open en gesloten tableau

Elke tak in een tableau correspondeert met een mogelijke route om een tegenvoorbeeld te vinden. Zo'n route kan op twee manieren eindigen:

a Er treedt eenzelfde formule links en rechts op in de sequent: we zeggen dan dat de tak van de boom *sluit*.

b Er treedt geen formule zowel links als rechts op, terwijl ook geen regel meer toepasbaar is. Dan noemen we de tak *open*.

We onderscheiden dan ook twee soorten tableaus:

1 Alle takken sluiten. Het tableau heet dan *gesloten*.

2 Minstens één tak blijft open. Het tableau heet dan *open*.

Deze twee gevallen corresponderen met geldigheid respectievelijk ongeldigheid van de met de topsequent corresponderende gevolgtrekking. We geven eerst nog eens een voorbeeld van het eerste geval:

Voorbeeld 3.5

Om te bepalen of de gevolgtrekking $p \vee q, p \rightarrow r, q \rightarrow s \ / \ r \vee s$ geldig is, maken we een semantisch tableau met behulp van de reductieregels:

$$p \vee q, p \rightarrow r, q \rightarrow s \circ r \vee s$$

1

$$p \vee q, p \rightarrow r, q \rightarrow s \circ r, s$$

2

$$r, p \vee q, q \rightarrow s \underline{\underline{\circ}} r, s \qquad p \vee q, q \rightarrow s \circ r, s, p$$

3

$$p, q \rightarrow s \underline{\underline{\circ}} r, s, p \qquad q, q \rightarrow s \circ r, s, p$$

4

$$s, q \underline{\underline{\circ}} r, s, p \qquad q \underline{\underline{\circ}} r, s, p, q$$

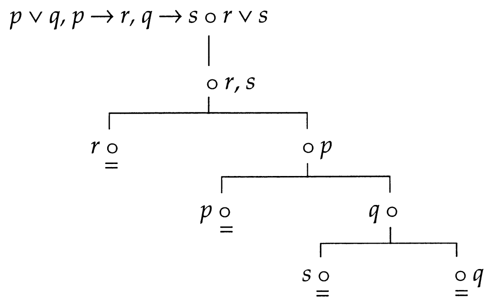
We hebben dus vier keer een reductieregel toegepast.
1 $\vee_R$ op $r \vee s$; dit levert de sequent bij 2.
2 $\rightarrow_L$ op $p \rightarrow r$; dit levert twee sequenten.
In de linker sequent komt de formule r links en rechts van $\circ$ voor. Dit geven we aan door het symbool $\circ$ dubbel te onderstrepen. Verder reduceren heeft dan volgens a geen zin, want in elke sequent die hierbij ontstaat, zal r links en rechts van $\circ$ staan. Een tegenvoorbeeld maakt de formules links van $\circ$ waar en rechts van $\circ$ onwaar. Omdat r zowel links als rechts voorkomt, en niet tegelijk waar en onwaar kan zijn, is er dus geen tegenvoorbeeld te vinden. De andere sequent staat bij 3.
3 $\vee_L$ op $p \vee q$; ook deze regel levert weer twee sequenten en ook hier geldt dat de linker sequent nooit een tegenvoorbeeld op zal leveren. De andere sequent staat bij 4.
4 $\rightarrow_L$ op $q \rightarrow s$; en ook deze regel levert twee sequenten. Beide sequenten zijn eindsequenten en kunnen niet verder gereduceerd worden. Bovendien komt in beide sequenten een formule aan weerszijden van $\circ$ voor, waardoor ook deze sequenten geen tegenvoorbeeld leveren.
Hiermee is het tableau klaar. Er is geen tegenvoorbeeld gevonden en de gevolgtrekking is dus geldig.

Notatie

We kunnen de notatie bij de knopen vereenvoudigen en de omvang van de boom beperken. De omvang van de boom beperken we door een sequent niet verder te reduceren als daarin een formule zowel links als rechts voorkomt: iedere tak door deze knoop sluit dan immers. Dit hebben we in het voorgaande voorbeeld reeds toegepast.
Tevens kunnen we bij de knopen van een semantisch tableau de schrijfwijze vereenvoudigen door slechts aan te geven wat er *verandert* in een sequent als deze gereduceerd wordt, zoals in het nu volgende voorbeeld.

Voorbeeld 3.6

Beschouw nogmaals voorbeeld 3.5. In de eerste stap reduceren we $r \vee s$ tot r, s, en alleen dat geven we aan. Bij de volgende stap reduceren we $p \rightarrow r$ volgens de regel $\rightarrow_L$, en we geven aan wat er met p en r gebeurt, enzovoorts. Het tableau wordt dan:

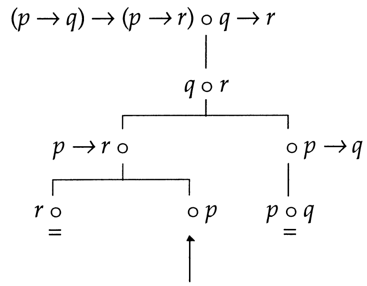


Een tak sluit als een formule zowel in een van zijn knopen links als in een van zijn (andere) knopen rechts van ∘ voorkomt.

Vervolgens geven we twee voorbeelden van een open tableau.

Voorbeeld 3.7

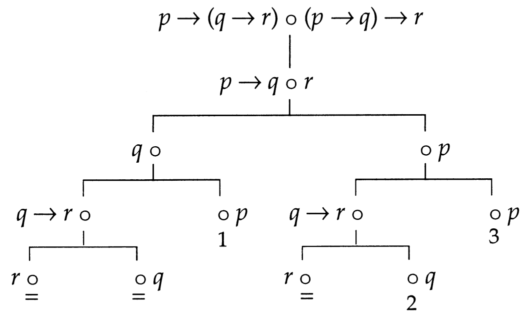
Om te testen of $(p \rightarrow q) \rightarrow (p \rightarrow r) \ / \ q \rightarrow r$ een geldige gevolgtrekking is, wordt het volgende tableau geconstrueerd:



Dit tableau bevat dus een open tak, aangegeven door de pijl. Een tegenvoorbeeld kan hiervan nu worden afgelezen door alle atomen die links van ∘ op deze tak voorkomen, waar te maken, en alle atomen die rechts van ∘ op de tak voorkomen, onwaar. Dus: $V(q) = 1$ en $V(r) = V(p) = 0$. Onder deze waardering geldt dan inderdaad ook $V((p \rightarrow q) \rightarrow (p \rightarrow r)) = 1$ en $V(q \rightarrow r) = 0$, zoals eenvoudig is na te gaan, ofwel $(p \rightarrow q) \rightarrow (p \rightarrow r) \not\models q \rightarrow r$.

Voorbeeld 3.8

De gevolgtrekking $p \rightarrow (q \rightarrow r)$ / $(p \rightarrow q) \rightarrow r$ is niet geldig (dit betekent: '$\rightarrow$ is niet associatief'). Met het volgende tableau vinden we drie tegenvoorbeelden, met cijfers aangegeven:



De drie tegenvoorbeelden zijn dus:
$V_1(r) = V_1(p) = 0$, $V_1(q) = 1$
$V_2(r) = V_2(p) = V_2(q) = 0$
$V_3(r) = V_3(p) = 0$, $V_3(q)$ willekeurig (V_3 'omvat' dus V_1 en V_2).
Merk op dat:

- sommige takken tegenvoorbeelden opleveren zonder waarheidswaarden voor elk atoom te specificeren;
- verschillende takken niet noodzakelijk verschillende tegenvoorbeelden hoeven te dragen.

3.3 CONSISTENTIE

Semantische tableaus zijn ook voor andere doeleinden bruikbaar dan geldigheidstests. Bijvoorbeeld, we kunnen ons de vraag stellen of een gegeven verzameling formules wel modellen heeft.

DEFINITIE 3.2

Semantisch consistent
Een formuleverzameling Σ is (*semantisch*) *consistent* als Σ een model heeft. We zeggen ook dat Σ *vervulbaar* is.

Inconsistentie

Een formuleverzameling die niet consistent is, heet *inconsistent*. Inconsistentie van een verzameling Γ is vaak ongewenst, omdat afwezigheid van modellen betekent dat uit die verzameling iedere formule geldig volgt, en daarmee ook de negatie ervan. De implicatie 'als V een model van Γ is, dan is het een model van φ' (ofwel $\Gamma \vDash \varphi$) is immers waar wegens het onwaar zijn van de aanname.

Consistentie testen met een tableau

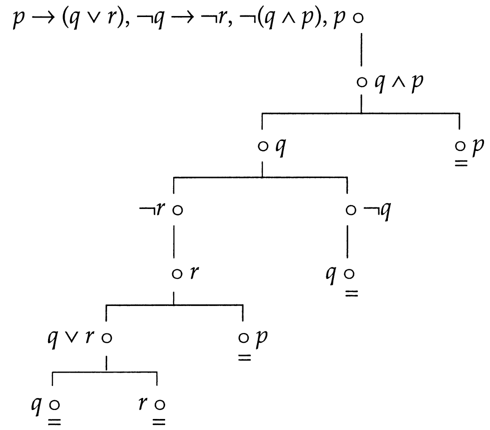
Consistentie van $\Sigma = \{\varphi_1, ..., \varphi_n\}$ valt nu als volgt met tableaus te testen. Beschouw de volgende sequent met rechts een lege rij:

$$\varphi_1, ..., \varphi_n \circ$$

Deze sequent heeft een tegenvoorbeeld als er een waardering V is die alle formules links van $\circ$ waar maakt en alle formules rechts van $\circ$ onwaar. Maar dat is gewoon een waardering die $\varphi_1, ..., \varphi_n$ waar maakt. Dus $\{\varphi_1, ..., \varphi_n\}$ is consistent als het tableau voor $\varphi_1, ..., \varphi_n \circ$ open is.

Voorbeeld 3.9

Is $\{p \to (q \vee r), \neg q \to \neg r, \neg(q \wedge p), p\}$ consistent?
Beschouw het volgende tableau:



Dit tableau sluit, dus de verzameling $\{p \to (q \vee r), \neg q \to \neg r, \neg(q \wedge p), p\}$ is inconsistent.

Tautologie testen met een tableau

Met behulp van een tableau kunnen we ten slotte ook testen of een gegeven formule een *tautologie* is. Om te controleren of een formule φ onder elke waardering waar is, bekijken we nu een sequent met aan de linkerkant het lege rijtje:

$$\circ\ \varphi$$

Als deze sequent geen tegenvoorbeeld heeft, dan is φ waar in elk model van de lege verzameling. Aangezien elke waardering model is van de lege verzameling, is elke waardering model van φ. Dus φ is een tautologie als een tableau van $\circ\ \varphi$ sluit.

Voorbeeld 3.10

Het volgende tableau sluit, er is dus geen waardering V waarvoor $V(\neg(p \wedge \neg p)) = 0$, en daarmee is $\neg(p \wedge \neg p)$ een tautologie.

$$\circ\ \neg(p \wedge \neg p)$$
$$p \wedge \neg p\ \circ$$
$$p, \neg p\ \circ$$
$$p \underset{=}{\circ} p$$

3.4 ADEQUAATHEID

Semantische tableaus werden geïntroduceerd als een methode om de geldigheid van een gevolgtrekking te bepalen. Een direct verband tussen deze twee noties wordt gegeven in de volgende stelling, die een verband legt tussen onze eerdere abstracte semantische notie van geldigheid en een meer concrete combinatorische:

STELLING 3.1

Adequaatheidsstelling
$\varphi_1, \ldots, \varphi_n \models \psi \Leftrightarrow$ er bestaat een gesloten tableau voor $\varphi_1, \ldots, \varphi_n \circ \psi$.

Deze stelling zegt dat de semantische tableaus een veilige methode vormen om te bepalen of een formule ψ een logisch gevolg is van $\varphi_1, \ldots, \varphi_n$. Een gegeven sequent kan meerdere tableaus hebben, afhankelijk van de volgorde waarin de reductieregels toegepast worden. De adequaatheidsstelling zegt onder meer dat alle mogelijke tableaus voor een sequent sluiten als ook maar één van hen sluit. Een bewijs hiervan wordt gegeven in hoofdstuk 5.

3.5 OPGAVEN

3.1 Stel $\varphi, \psi \models \alpha$, $\beta \models \gamma$ en $\psi, \alpha, \gamma \models \chi$. Indien nu bovendien bekend wordt dat χ onwaar is, maar ψ en β waar, wat weet u dan over φ?

3.2 Toon aan: $\varphi_1, \varphi_2 \models \psi \Leftrightarrow \varphi_1 \models \varphi_2 \to \psi \Leftrightarrow \models (\varphi_1 \wedge \varphi_2) \to \psi$.

3.3 Test met een semantisch tableau of de volgende equivalenties tautologieën zijn (als dat niet zo is, geef de tegenvoorbeelden):
i $(p \wedge (q \vee r)) \leftrightarrow ((p \wedge q) \vee (p \wedge r))$
ii $(p \to (q \to r)) \leftrightarrow ((p \to q) \to (p \to r))$

3.4 a Bewijs met een tableau de zogenaamde 'Wet van Hauber', die onder bepaalde voorwaarden omkeren van implicatie toestaat:
$p_1 \to q_1, p_2 \to q_2, p_1 \vee p_2, \neg(q_1 \wedge q_2) \models (q_1 \to p_1) \wedge (q_2 \to p_2)$

b Laat zien dat de volgende gevolgtrekkingen niet geldig zijn:
i $p \vee (q \wedge r) \,/\, (p \vee q) \wedge r$
ii $(p \wedge q) \vee r \,/\, p \wedge (q \vee r)$

3.5 Geef een voorbeeld van een geldige sequent met twee bijbehorende gesloten tableaus die in aantallen knopen verschillen. (Kunt u een meest efficiënte zoekstrategie bedenken om steeds aan het kleinste gesloten tableau te komen?)

* 3.6 Bewijs dat indien er gesloten tableaus bestaan voor de sequenten $\varphi \circ \psi$ en $\psi \circ \Sigma$, dan ook voor $\varphi \circ \Sigma$. (Dit is rechtstreeks niet zo eenvoudig te bewijzen!)

* 3.7 Beredeneer of er ook geldige gevolgtrekkingen kunnen zijn die geldig blijven als u de erin voorkomende connectieven door willekeurige andere vervangt.

Inhoud hoofdstuk 4

Propositielogica: afleidingen

Hoofdstuk 4

Propositielogica: afleidingen

4.1 INLEIDING: STRUCTUUR VAN REDENERINGEN

In het vorige hoofdstuk is een *semantische* kijk op geldigheid van een gevolgtrekking gegeven, in termen van modellen. In dit hoofdstuk zullen we logische geldigheid op een *syntactische* manier benaderen. Dit houdt in dat we ons zullen buigen over de vraag hoe conclusies getrokken kunnen worden uit een aantal aannames via een *redenering*. Dit noemen we het *bewijzen* of *afleiden* van de conclusie.

Stel eens dat we de volgende informatie hebben.
1 Jan vertelt een verhaal en Piet leest de krant.
2 Als Jan een verhaal vertelt, dan lacht Marie.
3 Als Piet de krant leest, dan kijkt Wilma televisie.

Uit deze aannames kunnen verschillende conclusies getrokken worden. Bovendien kan zo'n conclusie samen met een van de aannames of met een andere aanname weer een nieuwe conclusie opleveren.
Op deze manier kunnen we een redenering opzetten die bestaat uit kleine redeneerstappen:

stap 1: uit 1 concluderen we:
1a Jan vertelt een verhaal
Want, als Jan een verhaal vertelt en Piet de krant leest, dan is het zeker zo dat Jan een verhaal vertelt.
stap 2: uit 2 en 1a concluderen we:
2a Marie lacht
Want, volgens 1a vertelt Jan een verhaal, en 2 zegt dat als Jan een verhaal vertelt, dan lacht Marie.
stap 3: uit 1 concluderen we:
1b Piet leest de krant
Om ongeveer dezelfde reden als in stap 1.
stap 4: uit 3 en 1b concluderen we:
2b Wilma kijkt televisie

Om dezelfde reden als in stap 2.
stap 5: uit 2a en 2b concluderen we:
4 Marie lacht en Wilma kijkt televisie
We hebben geconcludeerd dat Marie lacht en we hebben geconcludeerd dat Wilma televisie kijkt. Maar dan mogen we ook concluderen dat Marie lacht en Wilma televisie kijkt.

Afleiding

De stappen 1 tot en met 5 bij elkaar noemen we een *afleiding*. In zo'n afleiding mag een aanname of een bereikte conclusie meerdere keren gebruikt worden. Aanname 1 bijvoorbeeld wordt zowel in stap 1 als in stap 3 gebruikt. (Overigens zijn onlangs binnen de informatica systemen van logisch bewijzen voorgesteld waarin aannames *niet* meerdere keren gebruikt mogen worden; hierover meer in hoofdstuk 21.)
De wijze waarop de afleiding hier is weergegeven is *lineair*, dat wil zeggen, de stappen worden na elkaar uitgevoerd. Bij langere en meer complexe afleidingen kunnen we de leesbaarheid sterk bevorderen door over te gaan op de tweedimensionale *boomvorm*. De kleinste eenheden in de opbouw daarvan zien er als volgt uit:

$$\frac{\varphi_1 \quad \cdots \quad \varphi_n}{\psi}$$

Hierbij wordt een conclusie ψ getrokken uit een aantal formules $\varphi_1, \ldots, \varphi_n$ die hetzij aannames zijn uit een gegeven beginverzameling, hetzij daaruit bereikte conclusies. Over het algemeen zullen er slechts één of twee formules boven de streep staan.
Laat bijvoorbeeld j, p, m en w propositieletters zijn die staan voor respectievelijk 'Jan vertelt een verhaal', 'Piet leest de krant', 'Marie lacht' en 'Wilma kijkt televisie'. Stap 1 in de eerdere redenering kan dan als volgt worden weergegeven:

$$\frac{j \wedge p}{j}$$

Stap 2 wordt:

$$\frac{j \qquad j \rightarrow m}{m}$$

Hier is de conclusie j uit stap 1 gebruikt om samen met aanname $j \to m$ de conclusie m op te leveren.
De stappen 1 en 2 kunnen nu op de volgende manier gecombineerd worden:

2a: $$\dfrac{\dfrac{j \wedge p}{j} \qquad j \to m}{m}$$

Dit schema is dus een afleiding van conclusie m uit aannames $j \wedge p$ en $j \to m$. De stappen 3, 4 en 5 worden in de nieuwe weergave:

3: $\dfrac{j \wedge p}{p}$ 4: $\dfrac{p \qquad p \to w}{w}$ 5: $\dfrac{m \qquad w}{m \wedge w}$

Stappen 3 en 4 kunnen net als 1 en 2 gecombineerd worden:

2b: $$\dfrac{\dfrac{j \wedge p}{p} \qquad p \to w}{w}$$

In stap 5 worden de conclusies 2a en 2b gebruikt om te concluderen tot $m \wedge w$. De bovengenoemde afleidingen 2a en 2b kunnen dus ook gecombineerd worden:

$$\dfrac{\dfrac{\dfrac{j \wedge p}{j} \qquad j \to m}{m} \qquad \dfrac{\dfrac{j \wedge p}{p} \qquad p \to w}{w}}{m \wedge w}$$

Hier staat precies beschreven hoe conclusie $m \wedge w$ is afgeleid uit de aannames $j \wedge p$, $j \to m$ en $p \to w$.

De conclusie in een afleiding staat altijd onderaan, terwijl aannames altijd bovenaan staan. Ze zijn immers niet afgeleid uit andere zinnen. De elementaire stappen zijn instructies om connectieven te gebruiken of in te voeren:

a $\dfrac{\varphi \wedge \psi}{\varphi}$

b $\dfrac{\varphi \wedge \psi}{\psi}$

c $\dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}$

d $\dfrac{\varphi \qquad \varphi \rightarrow \psi}{\psi}$

Dit idee gaan we nu generaliseren in het zogenaamde systeem van *natuurlijke deductie,* dat zijn naam ontleent aan de pretentie goed aan te sluiten bij hoe wij feitelijk redeneren.

4.2 NATUURLIJKE DEDUCTIE

Afleidingsregels

De genoemde stappen a, b, c, d en ook andere stappen die we later zullen tegenkomen, heten *afleidingsregels.* Deze kunnen in het algemeen optreden binnen een bewijs waar de formules boven de streep zelf reeds een bewijsboom boven zich hebben. Dit zien we dan ook terug in de algemene formulering van afleidingsregels. De algemene formulering van stappen a en b zegt hoe we informatie aan een conjunctie kunnen onttrekken:

$\wedge$–Eliminatieregels

Notatie: $\wedge$E

$$\dfrac{\begin{array}{c}\Sigma\\ \vdots\\ \varphi \wedge \psi\end{array}}{\varphi}\,\wedge\mathrm{E} \qquad\qquad \dfrac{\begin{array}{c}\Sigma\\ \vdots\\ \varphi \wedge \psi\end{array}}{\psi}\,\wedge\mathrm{E}$$

Hiermee wordt bedoeld: Een afleiding van $\varphi \wedge \psi$ met aannames uit formuleverzameling Σ, wordt een afleiding van φ, of een van ψ, uit Σ, door er de aangegeven stap onder te plaatsen. Omdat het connectief $\wedge$ in deze stap verdwijnt, spreekt men hier van een *eliminatieregel* (ook wel *gebruiksregel*). Bij beide afleidingsregels en ook alle volgende mag de verzameling aannames leeg zijn. Evenals we dat bij de overige regels zullen doen, schrijven we de naam van de regel rechts van de streep.

Regel c geeft het kenmerkende stramien waarmee wij in een redenering een conjunctie aantonen:

$\wedge$–Introductieregel

Notatie: $\wedge$I.

$$\dfrac{\begin{matrix}\Phi & \quad & \Psi \\ \vdots & & \vdots \\ \varphi & & \psi\end{matrix}}{\varphi \wedge \psi}\,\wedge\mathrm{I}$$

Dit betekent: een afleiding van φ uit Φ en een afleiding van ψ uit Ψ, verbonden door de aangegeven laatste stap, vormen een afleiding van $\varphi \wedge \psi$ uit de verenigde verzameling aannames $\Phi \cup \Psi$. Deze regel heet een *introductieregel* voor $\wedge$, omdat $\wedge$ boven de streep nog niet voorkwam, maar in de conclusie wel.
Met behulp van $\wedge$E en $\wedge$I kunnen nu afleidingen gemaakt worden:

Voorbeeld 4.1

Een afleiding van $(p \wedge q) \wedge r$ uit $p \wedge (q \wedge r)$ is als volgt:

$$\dfrac{\dfrac{\dfrac{p \wedge (q \wedge r)}{p}\,\wedge\mathrm{E} \qquad \dfrac{\dfrac{p \wedge (q \wedge r)}{q \wedge r}\,\wedge\mathrm{E}}{q}\,\wedge\mathrm{E}}{p \wedge q}\,\wedge\mathrm{I} \qquad \dfrac{\dfrac{p \wedge (q \wedge r)}{q \wedge r}\,\wedge\mathrm{E}}{r}\,\wedge\mathrm{E}}{(p \wedge q) \wedge r}\,\wedge\mathrm{I}$$

Twee keer is dus $\wedge$I toegepast: om $p \wedge q$ af te leiden en om $(p \wedge q) \wedge r$ af te leiden. In alle andere stappen wordt $\wedge$E gebruikt.

We zullen nu ook voor de overige connectieven eliminatie– en introductieregels invoeren.

$\rightarrow$–Eliminatieregel

Met de eliminatieregel voor $\rightarrow$ is al eerder kennisgemaakt. De informatie in een implicatie kan gebruikt worden zodra we het antecedent (de formule links van de implicatiepijl $\rightarrow$) ter beschikking hebben:

$$\dfrac{\begin{array}{c}\Sigma\\ \vdots\\ \varphi \to \psi\end{array} \qquad \begin{array}{c}\Phi\\ \vdots\\ \varphi\end{array}}{\psi}\ \to\mathrm{E}$$

Een afleiding van $\varphi \to \psi$ uit Σ en een afleiding van φ uit Φ combineren op de aangegeven manier tot een afleiding van ψ uit $\Sigma \cup \Phi$.

→-Introductieregel

Bij de introductieregel voor → komt iets nieuws naar voren, namelijk het intrekken van een aanname tijdens het construeren van een bewijs. Wanneer we een conclusie van de vorm $\varphi \to \psi$ willen beargumenteren, dan zullen we vaak φ als een tijdelijke 'hulpaanname' hanteren, die in de laatste stap van het bewijs wordt ingetrokken:

$$\dfrac{\begin{array}{c}\Sigma, \varphi\\ \vdots\\ \psi\end{array}}{\varphi \to \psi}\ \to\mathrm{I}, [-\varphi]$$

Wat dit zegt is het volgende. Gegeven een afleiding van ψ uit $\Sigma \cup \varphi$, hebben we met de laatste stap →I een afleiding van $\varphi \to \psi$ uit Σ alleen geconstrueerd: we trekken de aanname φ weer in. We staan ook toe om een aanname φ in te trekken die niet in de afleiding van ψ is gebruikt (een eenvoudige toepassing hiervan zult u hierna in voorbeeld 4.9 tegenkomen).

Vuistregels voor afleidingen

Met de tot nu toe genoemde regels kunnen we al meer interessante afleidingen maken. Anders dan bij eerdere testmethoden zoals waarheidstabellen en semantische tableaus, is het maken van een afleiding creatief. Enkele vuistregels hiervoor demonstreren we in het volgende voorbeeld.

Voorbeeld 4.2

Uit $p \to (q \to r)$ willen we afleiden $(p \to q) \to (p \to r)$. Hoe kunnen we dit aanpakken?

Van beneden naar boven

Om te beginnen analyseren we de gewenste conclusie. Dit betekent dat we van onder naar boven werken in de bewijsboom. De typische manier om zo'n implicatie af te leiden is, zoals hiervoor gezegd, door het antecedent $(p \to q)$ als hulpaanname te gebruiken, en dan samen met de hoofdaanname de conclusie $(p \to r)$ trachten af te leiden. En deze gedachte laat zich herhalen: het volstaat om een afleiding voor r te vinden met behulp van $p \to (q \to r)$, $(p \to q)$ en nog een hulpaanname p. Verdere analyse in deze richting is nu niet meer mogelijk.

Van boven naar beneden

We proberen nu de thans aanwezige aannames met behulp van eliminatieregels tot een bewijs van r in elkaar te passen. We werken dus van boven naar beneden in de bewijsboom. In ons speciale geval wijst de vorm van die aannames daartoe meteen de weg.

Samenvoegen

De resulterende bewijsboom:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{1}{p} \qquad \overset{2}{p \to q}}{q}\,{\to}\mathrm{E} \qquad \dfrac{\overset{1}{p} \qquad p \to (q \to r)}{q \to r}\,{\to}\mathrm{E}}{r}\,{\to}\mathrm{E}}{p \to r}\,{\to}\mathrm{I}, [-1]}{(p \to q) \to (p \to r)}\,{\to}\mathrm{I}, [-2]$$

We hebben de ingetrokken aannames op voor de hand liggende wijze genummerd. Dit zullen we voortaan meestal zo noteren.
Merk verder op hoe de bewijsboom in dit geval valt op te splitsen in een soort 'zandlopervorm': er is een eenvoudigste geval in het midden (te weten de formule r), waar van bovenaf naartoe wordt gewerkt met eliminatieregels, en van waaruit dan de conclusie wordt bereikt via introductieregels. Hoewel te simpel in het algemeen, is dit toch een nuttige manier om na te denken over gewenste bewijsstructuren.

Voorbeeld 4.3

Afleidingsregels voor $\wedge$ en $\to$ werken samen volgens hetzelfde stramien in een afleiding van $(p \wedge q) \to r$ uit $p \to (q \to r)$:

$$\dfrac{\dfrac{\dfrac{\overset{1}{p \wedge q}}{q}\,\wedge\mathrm{E} \qquad \dfrac{\dfrac{\overset{1}{p \wedge q}}{p}\,\wedge\mathrm{E} \qquad p \to (q \to r)}{q \to r}\,{\to}\mathrm{E}}{r}\,{\to}\mathrm{E}}{(p \wedge q) \to r}\,{\to}\mathrm{I}, [-1]$$

Ook hier geldt weer: de conclusie is niet meer afhankelijk van $p \wedge q$ omdat deze aanname is ingetrokken.

Expliciete aannames

Omdat aannames zo belangrijk zijn in bewijzen, geven we nog eens expliciet aan hoe de afhankelijkheden liggen voor elke knoop in de natuurlijke deductie van voorbeeld 4.2, door annotatie met accolades:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{1}{p\ \{p\}} \quad \overset{2}{p \to q\ \{p \to q\}}}{q\ \{p, p \to q\}}\ {\to}\mathrm{E} \qquad \dfrac{\overset{1}{p\ \{p\}} \quad p \to (q \to r)\ \{p \to (q \to r)\}}{q \to r\ \{p, p \to (q \to r)\}}\ {\to}\mathrm{E}}{r\ \{p, p \to q, p \to (q \to r)\}}\ {\to}\mathrm{E}}{p \to r\ \{p \to q, p \to (q \to r)\}}\ {\to}\mathrm{I}, [-1]}{(p \to q) \to (p \to r)\ \{p \to (q \to r)\}}\ {\to}\mathrm{I}, [-2]$$

Wat we nog eens zien zijn de volgende overervingen van aannames. Een aanname is alleen van zichzelf afhankelijk. De conclusie in een stap erft alle aannames van de formule(s) die boven de streep staan, behalve diegene die in die stap expliciet worden ingetrokken.
In het vervolg zullen we aannames alleen expliciet vermelden waar gevaar voor verwarring dreigt.

We gaan nu over naar de resterende connectieven, waarbij we iets compacter zullen formuleren.

∨–Introductieregel

Voor ∨ zijn er twee introductieregels, die de simpelste manier geven om een disjunctie te staven.

$$\dfrac{\begin{array}{c}\Sigma \\ \vdots \\ \varphi\end{array}}{\varphi \vee \psi}\ \vee\mathrm{I} \qquad \dfrac{\begin{array}{c}\Sigma \\ \vdots \\ \psi\end{array}}{\varphi \vee \psi}\ \vee\mathrm{I}$$

Voorbeeld 4.4

Een afleiding van $(p \to r) \wedge (q \to r)$ uit $(p \vee q) \to r$ ziet er zo uit:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{1}{p}}{p \vee q}\ \vee\mathrm{I} \quad (p \vee q) \to r}{r}\ {\to}\mathrm{E}}{p \to r}\ {\to}\mathrm{I}, [-1] \qquad \dfrac{\dfrac{\dfrac{\overset{2}{q}}{p \vee q}\ \vee\mathrm{I} \quad (p \vee q) \to r}{r}\ {\to}\mathrm{E}}{q \to r}\ {\to}\mathrm{I}, [-2]}{(p \to r) \wedge (q \to r)}\ \wedge\mathrm{I}$$

∨–Eliminatieregel

De eliminatieregel voor ∨ ligt iets moeilijker, omdat niet zonder meer duidelijk is hoe we in een redenering de informatie in een aanwezige disjunctie effectief kunnen gebruiken. Het meest 'natuurlijk' is het volgende principe van 'gevalsonderscheiding':

$$\dfrac{\begin{array}{c}\Sigma\\ \vdots\\ \varphi\vee\psi\end{array}\qquad \begin{array}{c}\Phi,\varphi\\ \vdots\\ \alpha\end{array}\qquad \begin{array}{c}\Psi,\psi\\ \vdots\\ \alpha\end{array}}{\alpha}\ \vee\text{E}, [-\varphi, -\psi]$$

Wat gebeurt hier? Stel dat er reeds een afleiding is van $\varphi \vee \psi$ uit Σ. Hieruit kunnen we niet direct tot een der disjuncten concluderen. Maar stel nu dat een zekere formule α in beide gevallen afleidbaar blijkt: zowel uit $\Phi \cup \varphi$ als uit $\Psi \cup \psi$. Dan beschouwen we de hier gegeven bewijsstructuur als een afleiding van α uit $\Sigma \cup \Phi \cup \Psi$. De aanwezigheid van de disjunctie links staat ons daarbij toe de hulpaannames φ en ψ in deze laatste stap in te trekken.

Voorbeeld 4.5

De regel ∨E wordt gebruikt in de volgende afleiding van $(p \vee q) \rightarrow r$ uit $(p \rightarrow r) \wedge (q \rightarrow r)$:

$$\dfrac{\dfrac{\overset{3}{p \vee q}\qquad \dfrac{\overset{1}{p}\qquad \dfrac{(p\rightarrow r)\wedge(q\rightarrow r)}{p\rightarrow r}\,\wedge\text{E}}{r}\,\rightarrow\text{E}\qquad \dfrac{\overset{2}{q}\qquad \dfrac{(p\rightarrow r)\wedge(q\rightarrow r)}{q\rightarrow r}\,\wedge\text{E}}{r}\,\rightarrow\text{E}}{r}\,\vee\text{E}, [-1, -2]}{(p\vee q)\rightarrow r}\,\rightarrow\text{I}, [-3]$$

Ten slotte geven we de bewijsregels voor negatie. Deze regels zijn op het eerste gezicht wat moeilijker dan men zou verwachten op grond van de eenvoud van negatie in onze eerdere testmethoden. De reden is dat negaties in redeneren een tamelijk speciale rol spelen: ze kondigen vaak 'rolwisselingen' aan, bijvoorbeeld van 'bewijzen' naar 'weerleggen'. Maar ook dit laatste proces verdient studie.

¬-Eliminatieregel

Als eerste geven we een eliminatieregel voor negatie, die ruwweg uitdrukt dat 'uit een tegenspraak alles volgt wat je maar wilt':

$$\begin{array}{cc} \Phi & \Psi \\ \vdots & \vdots \\ \varphi & \neg\varphi \\ \hline \multicolumn{2}{c}{\psi} \end{array}\ \neg\mathrm{E}$$

Uitleg: met een afleiding van φ uit Φ en een van $\neg\varphi$ uit Ψ, mogen we elke willekeurige conclusie ψ trekken uit de vereniging $\Phi \cup \Psi$.

¬-Introductieregel

De meest voor de hand liggende manier om een negatieve conclusie $\neg\psi$ te bewijzen is het weerleggen van ψ. Dit doen we in de introductieregel voor negatie. Het weerleggen gebeurt net als in de ¬E–regel door een tegenstrijdigheid aan te tonen:

$$\begin{array}{cc} \Sigma, \psi & \Phi, \psi \\ \vdots & \vdots \\ \varphi & \neg\varphi \\ \hline \multicolumn{2}{c}{\neg\psi} \end{array}\ \neg\mathrm{I}, [-\psi]$$

Merk op dat we hier de 'weerlegde' aanname ψ ook inderdaad intrekken. Eigenlijk doorbreekt deze regel enigszins de systematiek van ons systeem, aangezien de negatie niet alleen in de conclusie wordt ingevoerd maar ook in de onmiddellijke aannames van de slotregel voorkomt.

Voorbeeld 4.6

Een implicatie $p \to q$ zegt dat p een 'voldoende voorwaarde' voor het optreden van q is. De tegenhanger hiervan is dat q een 'noodzakelijke voorwaarde' voor p is: als q *niet* optreedt, dan kan p dat evenmin. Dit heet het principe van contrapositie. Een afleiding hiervan:

$$\dfrac{\dfrac{\dfrac{\overset{1}{p} \qquad p \to q}{q}\ \to\mathrm{E} \qquad \overset{2}{\neg q}}{\neg p}\ \neg\mathrm{I}, [-1]}{\neg q \to \neg p}\ \to\mathrm{I}, [-2]$$

Notatie

Om de leesbaarheid van bewijsbomen te vergroten, zullen we in het vervolg vaak bij de stappen van het bewijs de namen van de toegepaste regels weglaten.

Voorbeeld 4.7

We kunnen ook de eerder genoemde wetten van De Morgan nu afleiden. De eerste wet was de tautologie:

$$(\neg p \wedge \neg q) \leftrightarrow \neg(p \vee q)$$

We hebben geen aparte bewijsregels voor equivalentie, maar behandelen equivalentie steeds als dubbele implicatie. Hier volgen de twee richtingen van de afleiding (en in beide gevallen laten we de laatste $\rightarrow$I–stap weg). De eerste vergt van de negatieregels alleen $\neg$I:

$$\dfrac{\dfrac{\dfrac{\overset{1}{p}}{p \vee q} \qquad \neg(p \vee q)}{\neg p}\;\neg\mathrm{I}, [-1] \qquad \dfrac{\dfrac{\overset{2}{q}}{p \vee q} \qquad \neg(p \vee q)}{\neg q}\;\neg\mathrm{I}, [-2]}{\neg p \wedge \neg q}$$

Voor de omgekeerde richting blijkt ook de $\neg$E–regel nodig:

$$\dfrac{\dfrac{\overset{3}{p \vee q} \qquad \dfrac{\overset{1}{p} \qquad \dfrac{\neg p \wedge \neg q}{\neg p}}{\neg(p \vee q)}\;\neg\mathrm{E} \qquad \dfrac{\overset{2}{q} \qquad \dfrac{\neg p \wedge \neg q}{\neg q}}{\neg(p \vee q)}\;\neg\mathrm{E}}{\neg(p \vee q)}\;\vee\mathrm{E}, [-1, -2] \qquad p \vee q}{\neg(p \vee q)}\;\neg\mathrm{I}, [-3]$$

De absurde bewering $\perp$

Men zou een negatie $\neg\varphi$ ook kunnen opvatten als een implicatie $\varphi \rightarrow \perp$ van φ naar een *'absurde' atomaire bewering* $\perp$ waaruit alles volgt. De absurde bewering $\perp$ wordt ook wel *falsum* genoemd. In dat geval is de $\neg$E–regel te vervangen door $\rightarrow$E, gevolgd door een '$\perp$–regel' die uit $\perp$ tot elke willekeurige formule concludeert. Evenzo kunnen we dan negaties introduceren via de oude regel $\rightarrow$I, zodat de $\neg$I–regel komt te vervallen.

Bewijs uit het ongerijmde
De $\neg E^*$–regel

Naast het bewijzen van negaties door weerlegging van een bewering, bestaat er een in de wetenschap veel gebruikte bewijsfiguur die juist 'uit het ongerijmde' werkt: een bewering wordt bewezen door haar negatie te weerleggen. Dit leidt tot een extra regel in ons systeem, die onafhankelijk is van de tot nu toe ingevoerde:

$$\dfrac{\begin{array}{c}\Sigma, \neg\psi \\ \vdots \\ \varphi\end{array} \qquad \begin{array}{c}\Phi, \neg\psi \\ \vdots \\ \neg\varphi\end{array}}{\psi}\ \neg\mathrm{E}^*, [-\neg\psi]$$

Let op het intrekken van de aanname $\neg\psi$. We kunnen dit desgewenst ook een 'eliminatieregel' noemen.

Voorbeeld 4.8

Een direct gevolg (en zelfs een equivalent) van de nieuwe negatieregel is het 'principe van dubbele negatie' $\neg\neg p \leftrightarrow p$:

$$\dfrac{\dfrac{\overset{1}{\neg p} \qquad \overset{2}{\neg\neg p}}{p}\ \neg\mathrm{E}^*, [-1]}{\neg\neg p \to p}\ \to\mathrm{I}, [-2]$$

In de andere richting is dit met de eerste twee negatieregels afleidbaar.

Het is een subtiele kwestie om te bepalen welke negatieprincipes echt nodig zijn om propositielogische principes te bewijzen. Een instructief voorbeeld is de tweede Wet van De Morgan, die veel lijkt op de eerste:

$$\neg (p \wedge q) \leftrightarrow (\neg p \vee \neg q)$$

Met de oorspronkelijke twee negatieregels is echter alleen $\neg (p \wedge q)$ af te leiden uit $\neg p \vee \neg q$. Om ook $\neg p \vee \neg q$ te kunnen afleiden uit $\neg (p \wedge q)$, moet $\neg\mathrm{E}^*$ worden gebruikt:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\overset{3}{\neg(\neg p \vee \neg q)} \qquad \dfrac{\overset{1}{\neg p}}{\neg p \vee \neg q}}{p}\ \neg\mathrm{E}^*, [-1] \qquad \dfrac{\overset{3}{\neg(\neg p \vee \neg q)} \qquad \dfrac{\overset{2}{\neg q}}{\neg p \vee \neg q}}{q}\ \neg\mathrm{E}^*, [-2]}{p \wedge q} \qquad \overset{4}{\neg(p \wedge q)}}{\neg p \vee \neg q}\ \neg\mathrm{E}^*, [-3]}{\neg(p \wedge q) \rightarrow (\neg p \vee \neg q)}\ \rightarrow\mathrm{I}, [-4]
$$

Natuurlijke deductie

Alle regels die in deze paragraaf behandeld zijn, vormen tezamen het afleidingssysteem van de *natuurlijke deductie*. In het volgende overzicht staan ze bij elkaar:

$$
\dfrac{\begin{matrix}\Sigma\\ \vdots\\ \varphi \wedge \psi\end{matrix}}{\varphi}\ \wedge\mathrm{E} \qquad \dfrac{\begin{matrix}\Sigma\\ \vdots\\ \varphi \wedge \psi\end{matrix}}{\psi}\ \wedge\mathrm{E} \qquad\qquad \dfrac{\begin{matrix}\Sigma\\ \vdots\\ \varphi\end{matrix} \qquad \begin{matrix}\Phi\\ \vdots\\ \psi\end{matrix}}{\varphi \wedge \psi}\ \wedge\mathrm{I}
$$

$$
\dfrac{\begin{matrix}\Sigma\\ \vdots\\ \varphi \rightarrow \psi\end{matrix} \qquad \begin{matrix}\Phi\\ \vdots\\ \varphi\end{matrix}}{\psi}\ \rightarrow\mathrm{E} \qquad\qquad \dfrac{\begin{matrix}\Sigma, \varphi\\ \vdots\\ \psi\end{matrix}}{\varphi \rightarrow \psi}\ \rightarrow\mathrm{I}, [-\varphi]
$$

$$
\dfrac{\begin{matrix}\Sigma\\ \vdots\\ \varphi \vee \psi\end{matrix} \qquad \begin{matrix}\Phi, \varphi\\ \vdots\\ \alpha\end{matrix} \qquad \begin{matrix}\Psi, \psi\\ \vdots\\ \alpha\end{matrix}}{\alpha}\ \vee\mathrm{E}, [-\varphi, -\psi] \qquad \dfrac{\begin{matrix}\Sigma\\ \vdots\\ \varphi\end{matrix}}{\varphi \vee \psi}\ \vee\mathrm{I} \qquad \dfrac{\begin{matrix}\Sigma\\ \vdots\\ \psi\end{matrix}}{\varphi \vee \psi}\ \vee\mathrm{I}
$$

$$
\dfrac{\begin{matrix}\Sigma\\ \vdots\\ \varphi\end{matrix} \qquad \begin{matrix}\Phi\\ \vdots\\ \neg\varphi\end{matrix}}{\psi}\ \neg\mathrm{E} \qquad \dfrac{\begin{matrix}\Sigma, \neg\psi\\ \vdots\\ \varphi\end{matrix} \qquad \begin{matrix}\Phi, \neg\psi\\ \vdots\\ \neg\varphi\end{matrix}}{\psi}\ \neg\mathrm{E}^*, [-\neg\psi] \qquad \dfrac{\begin{matrix}\Sigma, \psi\\ \vdots\\ \varphi\end{matrix} \qquad \begin{matrix}\Phi, \psi\\ \vdots\\ \neg\varphi\end{matrix}}{\neg\psi}\ \neg\mathrm{I}, [-\psi]
$$

4.3 AFLEIDBAAR EN CONSISTENT

Afleidbaar

Nu we eenmaal een exact bewijssysteem hebben gepresenteerd, kunnen we ook preciezer een aantal belangrijke begrippen formuleren die vaak voorkomen in wiskunde en informatica. Het eerste daarvan is uiteraard het volgende:

DEFINITIE 4.1

Afleidbaar
Een formule φ heet *afleidbaar* uit een verzameling aannames Σ als er een afleiding van φ bestaat waarin aan het eind alleen nog aannames uit Σ van kracht zijn. Notatie: $\Sigma \vdash \varphi$.

Als φ afleidbaar is zonder aannames, dan heet φ een *stelling*. In dat geval is Σ leeg en noteren we:

$$\vdash \varphi$$

Als φ niet afleidbaar is uit Σ, noteren we: $\Sigma \nvdash \varphi$. Evenzo, als φ geen stelling is: $\nvdash \varphi$. Alle natuurlijke deducties leiden tot stellingen indien men alle aannames intrekt met de regel $\rightarrow$I. (Zo leidt bijvoorbeeld $\varphi, \psi \vdash \xi$ met $\rightarrow$I tot $\varphi \vdash \psi \rightarrow \xi$ en weer met $\rightarrow$I tot $\vdash \varphi \rightarrow (\psi \rightarrow \xi)$.)

Wellicht ten overvloede is hier nog een voorbeeld van een simpele, maar bekende stelling:

Voorbeeld 4.9

Een afleiding van $p \rightarrow (q \rightarrow p)$:

$$\dfrac{\dfrac{\overset{1}{p}}{q \rightarrow p}}{p \rightarrow (q \rightarrow p)}\,[-1]$$

Let op de 'lege intrekking' bij de eerste toepassing van $\rightarrow$I!

Consistent

Met het begrip afleidbaar kunnen we vervolgens syntactische consistentie definiëren:

DEFINITIE 4.2

Syntactisch consistent
Een verzameling formules Γ heet *syntactisch consistent* wanneer er geen formule φ is waarvoor zowel $\Gamma \vdash \varphi$ als $\Gamma \vdash \neg\varphi$.

We zullen in het vervolg kortweg de term 'consistent' gebruiken. Een vergelijking met de *semantische* notie van consistentie uit hoofdstuk 3 stellen we uit tot het volgende hoofdstuk. Een voorbeeld van een consistente verzameling is $\{\neg p, p \rightarrow q, q\}$.

Inconsistent

Een verzameling formules die niet consistent is heet *inconsistent*. Inconsistent is bijvoorbeeld de verzameling $\Gamma = \{p, p \rightarrow q, \neg q\}$. Er geldt namelijk $\Gamma \vdash \neg q$ en $\Gamma \vdash q$.

Het nauwe verband tussen consistentie en afleidbaarheid wordt nader uitgewerkt in de volgende resultaten. Deze worden hier overigens ook opgevoerd ter illustratie van het gebruik van logica voor het redeneren *over* bewijssystemen (zie paragraaf 5 van appendix 1), waarmee we al wat vooruitlopen op het volgende hoofdstuk over metaredeneren. Bij metaredeneringen gebruiken we het teken $\square$ om het einde van een bewijs aan te geven.

BEWERING 4.1

Een formuleverzameling Γ is consistent desda er bestaat een formule φ zodat $\Gamma \nvdash \varphi$.

Bewijs

$\Rightarrow$ Als Γ consistent is, dan is er geen formule φ waarvoor $\Gamma \vdash \varphi$ en $\Gamma \vdash \neg\varphi$. Voor een willekeurige formule ξ geldt dan $\Gamma \nvdash \xi$ of $\Gamma \nvdash \neg\xi$. Als $\Gamma \nvdash \xi$, dan is ξ de gezochte formule, als $\Gamma \nvdash \neg\xi$, dan $\neg\xi$.

$\Leftarrow$ (Contrapositie) Als Γ inconsistent is, dan bestaat er een formule ξ zodat $\Gamma \vdash \xi$ en $\Gamma \vdash \neg\xi$. Maar dan is met $\neg$E elke formule afleidbaar uit Γ, ofwel er bestaat geen formule φ met $\Gamma \nvdash \varphi$. $\square$

BEWERING 4.2

$\Gamma \nvdash \varphi \Leftrightarrow \Gamma \cup \{\neg\varphi\}$ is consistent.

Bewijs

We passen op zowel $\Rightarrow$ als $\Leftarrow$ contrapositie toe:

$\Rightarrow$ Stel $\Gamma \cup \{\neg\varphi\}$ is inconsistent. Dan is er een formule ξ zodat $\Gamma \cup \{\neg\varphi\} \vdash \xi$ en $\Gamma \cup \{\neg\varphi\} \vdash \neg\xi$. De eliminatieregel $\neg$E* geeft nu: $\Gamma \vdash \varphi$.

$\Leftarrow$ Als $\Gamma \vdash \varphi$, dan ook $\Gamma \cup \{\neg\varphi\} \vdash \varphi$. Omdat ook $\Gamma \cup \{\neg\varphi\} \vdash \neg\varphi$, is $\Gamma \cup \{\neg\varphi\}$ inconsistent. $\square$

In de wiskunde en informatica zijn we vaak niet zozeer geïnteresseerd in redeneren vanuit willekeurige stelsels aannames. We denken veeleer aan specifieke theorieën, gegeven door axioma's, of aan gegevensbanken met lijsten van opgeslagen informatie. Dit wordt behandeld in de nu volgende paragraaf.

4.4 AXIOMATISCH AFLEIDEN

Axioma

Natuurlijke deductie is niet de enige manier om afleidbaarheid vorm te geven. Terwijl in dit systeem de afleidingsregels de kern vormen, zijn er ook systemen waarin *axioma's* de hoofdrol spelen, meer zoals dat in de klassieke meetkunde het geval is. Een axioma is een formule die op elk moment in een bewijs gebruikt kan worden. Dit sluit meer aan bij de traditionele manier van 'axiomatisch bewijzen' in de wiskunde.

Axiomatiek

In het algemeen bestaat een axiomatisch systeem of *axiomatiek* uit een verzameling axioma's en afleidingsregels. Ter illustratie volgt hier de axiomatiek S, die bestaat uit een aantal axioma's plus een afleidingsregel voor propositielogisch redeneren. We beperken ons voor het gemak tot twee connectieven ($\{\rightarrow, \neg\}$ is functioneel volledig):

Axioma's:
a $\varphi \rightarrow (\psi \rightarrow \varphi)$
b $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$
c $(\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)$

Afleidingsregel ('Modus Ponens'):
d Uit φ en $(\varphi \rightarrow \psi)$ mogen we ook ψ afleiden.

Als met aannames uit een formuleverzameling Σ met behulp van de axioma's van S en Modus Ponens een formule ψ valt af te leiden, dan noteren we $\Sigma \vdash_S \psi$. Merk op dat $\vdash_S \varphi$ als φ een axioma is, dat $\Sigma \vdash_S \varphi$ als $\varphi \in \Sigma$, en dat Modus Ponens nu de vorm krijgt: uit $\Sigma \vdash_S \varphi$ en $\Sigma' \vdash_S \varphi \rightarrow \psi$ volgt $\Sigma \cup \Sigma' \vdash_S \psi$.

Deze twee benaderingen, hoe verschillend ook in de vorm van hun bewijsstructuren, blijken equivalent in de volgende zin:

STELLING 4.1

Als Σ een formuleverzameling is en φ een formule, dan geldt: $\Sigma \vdash \varphi$ in natuurlijke deductie $\Leftrightarrow$ $\Sigma \vdash_S \varphi$.

Uiteraard eisen we dat in φ slechts de connectieven $\rightarrow$ en $\neg$ gebruikt worden. We geven deze stelling zonder bewijs.

Het verschil in geaardheid tussen de twee bewijssystemen uit zich onder meer daarin dat een principe dat voor natuurlijke deductie een directe regel is, namelijk de $\rightarrow$I–regel, voor S een eigenschap is die bewezen moet worden. We geven deze eigenschap zonder bewijs:

BEWERING 4.3

'Deductiestelling'
Als $\varphi \vdash_S \psi$, dan $\vdash_S \varphi \to \psi$.

4.5 THEORETISCHE ASPECTEN VAN HET SYSTEEM VAN NATUURLIJKE DEDUCTIE

We bespreken nu nog enkele algemene kwesties die door het systeem van natuurlijke deductie worden opgeroepen.

Ten eerste zal duidelijk zijn dat onze bewijstheoretische analyse van propositielogische geldigheid een geheel ander 'logisch landschap' oplevert dan de eerdere waarheidstabellen of semantische tableaus. Met name suggereert de indeling in bewijsregels dat het ook zinvol zou kunnen zijn om bepaalde deelsystemen te beschouwen, ontstaan door hergroepering van mogelijke principes. Met afleidingsstelsels kan men dus 'experimenteren': een kenmerkend verschijnsel in de moderne logica.

Intuïtionistische logica

Een belangrijk deelsysteem van natuurlijke deductie is het volgende. Zonder de negatieregel $\neg$E* ontstaat een heel natuurlijke 'alternatieve logica', te weten de *intuïtionistische*. Hierin is bijvoorbeeld de regel voor dubbele negatie $\neg\neg p \to p$ geen stelling, en evenmin het zogenaamde 'principe van het uitgesloten derde' $p \vee \neg p$. De intuïtionistische logica is ontstaan in de grondslagen van de wiskunde, als meer passende logica voor 'constructief' redeneren. Het *weerleggen* van de negatie van een bewering φ in de negatieregel $\neg$E* is namelijk niet hetzelfde als het *construeren* van φ. Dit aspect lichten we verder toe bij de behandeling van de intuïtionistische logica in hoofdstuk 12.
De laatste tijd is de intuïtionistische logica ook binnen de informatica naar voren getreden, bijvoorbeeld op de gebieden van programmacorrectheid en programmasynthese. Men bewijst daar eerst met intuïtionistische natuurlijke deductie dat er oplossingen voor een probleem bestaan. Daarna worden uit deze bewijzen effectief algoritmen afgelezen om dat probleem op te lossen.

Beslisbaarheid

'Bewijzen zoeken' is in principe geen effectieve beslissingsmethode voor geldigheid. Je weet dat de onderzochte gevolgtrekking geldig is zodra je een bewijs produceert; maar hoe toon je in geval van onbewijsbaarheid na eindig veel stappen aan dat er, onder de oneindig veel mogelijkheden daartoe, helemaal geen bewijs voor die gevolgtrekking is? We zagen dit verschil met bijvoorbeeld de beslissingsmethode voor semantische tableaus van hoofdstuk 3 ook al daarin, dat bewijzen zoeken doorgaans enige creativiteit vergt. Alleen een gedetailleerder analyse kan ons nader uitsluitsel geven. Het blijkt dan dat zowel ons standaardsysteem van natuurlijke deductie als zijn intuïtionistische variant effectief beslisbaar zijn. Dit is echter geen ijzeren wet: in latere hoofdstukken zullen we logische systemen tegenkomen die niet beslisbaar zijn. In hoofdstuk 14 gaan we nader in op de begrippen beslisbaarheid en berekenbaarheid.

Automatisch stellingbewijzen

Vanuit de informatica bezien is het systeem van natuurlijke deductie een voorbeeld van een formele bewijstheorie die geprogrammeerd kan worden. Het is daardoor in principe geschikt als automatische 'theorem prover'. In hoofdstuk 16 bespreken we een andere opzet voor het programmeren van logische bewijssystemen, te weten 'resolutie'.

Volledigheid

Het meest urgent, gezien onze eerdere hoofdstukken, is het verband tussen syntactische afleidbaarheid en semantische geldigheid in bewijssystemen. Het is namelijk één ding om een natuurlijke–deductiecalculus voor afleidbaarheid $\vdash$ te hebben, maar een andere kwestie is hoe die notie zich verhoudt tot de oorspronkelijke semantische notie van geldigheid $\models$. Een zeer fraai en fundamenteel logisch resultaat is de zogenaamde volledigheidsstelling, die zegt dat deze zo verschillend geaarde benaderingen toch overeenstemmen:

STELLING 4.2

Volledigheid
Als Σ een formuleverzameling is en φ een formule, dan geldt:
$\Sigma \vdash \varphi \Leftrightarrow \Sigma \models \varphi$.

Er zitten twee kanten aan deze equivalentie, die ook meer in het algemeen aan logische systemen vallen te onderscheiden:

correctheid: afleidbare gevolgtrekkingen zijn semantisch geldig
volledigheid: semantisch geldige gevolgtrekkingen zijn afleidbaar.

Merk op dat 'volledigheid' zowel de naam van de stelling is, als van de implicatie $\Sigma \models \varphi \Rightarrow \Sigma \vdash \varphi$. Deze wellicht wat ongelukkige naamgeving is historisch gegroeid.
Het bewijs van deze stelling wordt in het volgende hoofdstuk gegeven.

4.6 OPGAVEN

4.1 Bewijs met natuurlijke deductie:
i $p \to (q \to r) \vdash q \to (p \to r)$
ii $(p \wedge q) \to r \vdash p \to (q \to r)$
iii $p \to q\,, p \to r \vdash p \to (q \wedge r)$, en de twee beweringen die uit de omkering voortkomen.

4.2 Bewijs met natuurlijke deductie:
i $\vdash p \to \neg\neg p$
ii $\neg p \vee q \vdash p \to q$, en omgekeerd
iii $\neg p \to \neg q \vdash q \to p$
Waar precies heeft u $\neg$E* nodig?

4.3 De wetten van De Morgan zijn twee van de zogenaamde *Boolese axioma's*. Bewijs ook de Boolese axioma's voor distributiviteit:
i $p \vee (q \wedge r) \vdash (p \vee q) \wedge (p \vee r)$, en omgekeerd
ii $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$, en omgekeerd
Bewijs de Boolese axioma's voor *absorptie*:
iii $p \vee (p \wedge q) \vdash p$, en omgekeerd
iv $p \wedge (p \vee q) \vdash p$, en omgekeerd

4.4 Een formule kan op meerdere plaatsen in een natuurlijke–deductieboom voorkomen. Omdat elk voorkomen van verschillende aannames afhankelijk kan zijn, mogen deze voorkomens niet met elkaar geïdentificeerd worden. Laat dit zien in een voorbeeld.

4.5 Toon zonder beroep op de volledigheidsstelling aan dat als $\varphi \vdash \psi$ en $\psi \vdash \chi$, dan $\varphi \vdash \chi$.

4.6 Geef introductie– en gebruiksregels voor de equivalentie $\leftrightarrow$.

4.7 Bewijs de volgende bewering:
Als een verzameling Σ consistent is, dan geldt voor alle formules φ en ψ: $\Sigma \cup \{\varphi \vee \psi\}$ is consistent desda $\Sigma \cup \{\varphi\}$ is consistent of $\Sigma \cup \{\psi\}$ is consistent.

* 4.8 Stel dat we in natuurlijke deductie voortaan alleen nog maar het intrekken toestaan van aannames die echt in bewijzen worden gebruikt. (Men spreekt dan wel van 'relevante deductie'.)
Geef enkele voorbeelden van voorheen afleidbare principes die dat dan niet meer zijn.

* 4.9 Bij logicatentamens in de Middeleeuwen speelden docent en student het zogenaamde 'Obligatio–spel'. De docent deed achtereenvolgens beweringen, die de student moest 'accepteren' (dan kwam de bewering op zijn conto) of 'verwerpen' (dan kwam de negatie op zijn conto). Wie na een zeker aantal ronden nog steeds een consistente verzameling op zijn conto had, was geslaagd. Laat zien dat er altijd een slaagmogelijkheid is, wat de docent ook voorstelt.

Inhoud hoofdstuk 5

Propositielogica: metatheorie

Hoofdstuk 5

Propositielogica: metatheorie

5.1 METATHEORIE

Metabeweringen

We hebben tot nu toe voornamelijk kennisgemaakt met de propositielogica als instrument om mee te werken, bijvoorbeeld door toepassen van methodes als waarheidstabellen, tableaus of natuurlijke deductie. In dit hoofdstuk stellen we ons op een 'hoger niveau', en beschouwen eigenschappen van dit systeem als zodanig. De resultaten die aldus worden verkregen, heten wel 'metabeweringen': dit zijn dus stellingen *over* het systeem, in tegenstelling tot de eerdere afleidbare formules, die optraden als 'stellingen' *binnen* het systeem van de propositielogica. Dit soort reflectie op het instrumentarium dat men gebruikt, is overigens zeer kenmerkend voor de moderne logica, waar vele beroemde resultaten samenhangen met de vraag wat de mogelijkheden en beperkingen van logische systemen zijn, en hoe deze onderling weer samenhangen.
Twee voorbeelden van dergelijke metabeweringen uit het voorafgaande zijn functionele volledigheid van connectieven en correctheid van bewijssystemen. We zullen ze nu nog eens precies formuleren en bewijzen. In sommige gevallen zal dat bewijzen niets anders zijn dan een precies onderbouwen van een voorgevoel dat we reeds in de praktijk hadden verworven, maar in andere gevallen gaat het om verrassende beweringen die historisch ook echt ontdekt moesten worden.

Bewijsvormen

Metastellingen doen algemene uitspraken over ons systeem van logische formules, tableaus en bewijzen. Wat voor bewijsmethoden zijn op dit metaniveau zelf geëigend?
Ten eerste blijkt natuurlijk dat het redeneren over ons logisch systeem met in principe dezelfde logica gebeurt. Hoewel dit in zekere zin circulair is, is hiervoor echter geen alternatief. Het redeneren is zowel object als middel van onderzoek: we hebben in de afgelopen hoofdstukken een fragment uit de bestaande praktijk van redeneren geformaliseerd, met gebruikmaking van uiteraard diezelfde

redeneertrant. Zo komt bijvoorbeeld de equivalentie van uitspraken '$\Leftrightarrow$' overeen met de propositielogische equivalentie '$\leftrightarrow$'. Eveneens zijn we bijvoorbeeld het principe van contrapositie, binnen de propositielogica geformuleerd, ook tegengekomen in het bewijs van bewering 4.2 over het verband tussen afleidbaarheid en consistentie.
Naast dit algemene gebruik van logica voor metaredeneren zijn er ook specifieke redeneervormen op metaniveau die de aandacht verdienen. Een belangrijke vorm van argumenteren die ons hier vaak van pas komt, is *inductie*: zowel in de gangbare wiskundige vorm zoals u dat principe reeds kent, als in een speciaal op logische objecten toegespitste vorm. Op deze laatste vorm gaan we nu nader in.

5.2 FORMULE–INDUCTIE

Een metabewering heeft vaak betrekking op alle formules. Omdat het nu eenmaal onmogelijk is om voor elke formule afzonderlijk na te gaan of de bewering klopt, moet er een andere vorm gevonden worden om metabeweringen te toetsen. Vaak lukt dat met zogenaamde *formule–inductie*. De basis hiervoor is de inductieve opbouw van formules die reeds in hoofdstuk 2 werd gebruikt: formules opbouwen vanuit de basisverzameling van atomen via eindige combinatie met logische operaties.
Parallel aan die inductieve opbouw bleken we algemene begrippen voor formules inzichtelijk inductief te kunnen definiëren via een basisafspraak plus recursie over de componenten. Voorbeelden hiervan zijn syntactische substitutie en semantische toekenning van waarheidswaarden. Evenzo blijkt het nu mogelijk om volgens dit patroon een algemene bewering B aan te tonen voor alle propositielogische formules:

Basisstap
Laat zien dat B opgaat voor de atomen.
Inductiestap
Laat zien dat, als B opgaat voor formules φ, ψ, dan ook voor hun combinaties $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ en $\varphi \leftrightarrow \psi$.

De aanname in de inductiestap noemen we daarbij de 'inductiehypothese'. Hoe dit werkt, zien we het beste aan een concreet voorbeeld.

Subformule

Een formule heet een *subformule* van een formule φ als die formule als een aaneengesloten deel in φ voorkomt. Er zijn acht subformules in bijvoorbeeld $((p \wedge q) \wedge (p \rightarrow \neg q))$: p (twee keer), q (ook twee keer), $\neg q$, $(p \wedge q)$, $(p \rightarrow \neg q)$ en $((p \wedge q) \wedge (p \rightarrow \neg q))$. Merk op dat elke formule een subformule van zichzelf is. Voor het gemak noteren we het aantal subformules dat in φ voorkomt als '$subf(\varphi)$'.

Lengte

De *lengte* van een formule φ ($lengte(\varphi)$) is het aantal symbolen in φ, zonder de haakjes mee te tellen. Dus, bijvoorbeeld, $lengte(((p \wedge q) \wedge (p \rightarrow \neg q))) = 8$.
We hebben nu het volgende eenvoudige verband:

BEWERING 5.1

Voor elke formule φ is het aantal voorkomens van subformules in φ gelijk aan de lengte van φ.

Bewijs

Basisstap

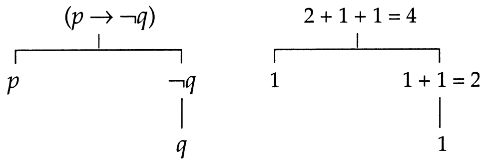
Voor een atomaire formule is het aantal subformules gelijk aan de lengte, want een atomaire formule heeft maar één voorkomen van een subformule, namelijk zichzelf, en ook de lengte van een atomaire formule is 1.

Inductiestap

Stel dat de bewering reeds geldt voor twee willekeurige formules φ en ψ: dat wil zeggen, $subf(\varphi) = lengte(\varphi)$ en $subf(\psi) = lengte(\psi)$. We moeten bewijzen dat de bewering dan ook opgaat voor $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ en $\varphi \leftrightarrow \psi$.
Geval 1: negatie. We hebben dan de volgende berekening: $subf(\neg\varphi) = subf(\varphi) + 1 = lengte(\varphi) + 1 = lengte(\neg\varphi)$.
De eerste gelijkheid is juist omdat $\neg\varphi$ één subformule meer heeft dan φ, namelijk $\neg\varphi$. Bij de tweede gelijkheid passen we de inductiehypothese $subf(\varphi) = lengte(\varphi)$ toe. De laatste gelijkheid geldt omdat in $\neg\varphi$ één symbool meer voorkomt dan in φ, namelijk $\neg$.
Geval 2: binaire connectieven. Het voorbeeld van conjunctie illustreert het algemene geval: $subf(\varphi \wedge \psi) = subf(\varphi) + subf(\psi) + 1$ (de conjunctie heeft als subformules die van haar componenten plus zichzelf) $= lengte(\varphi) + lengte(\psi) + 1$ (wegens de inductiehypothese) $= lengte(\varphi \wedge \psi)$ ($\wedge$ telt nu ook mee). □

Voorbeeld 5.1

In een concreet voorbeeld illustreren we dit bewijsproces nog eens. Lees van beneden naar boven en merk op dat bij de overgangen de inductiehypothese gebruikt wordt:



Net als natuurlijke inductie in de wiskunde, is formule–inductie geen wondermiddel om willekeurige resultaten zonder moeite te bewijzen.

Het is veeleer een overzichtelijke methode om onze redeneringen op te zetten, waarbij soms nog extra ideeën nodig zijn. We zullen dit nu illustreren met een reeks van metastellingen, die teruggrijpen op de eerdere hoofdstukken 2, 3 en 4.

5.3 FUNCTIONELE VOLLEDIGHEID

STELLING 5.1

Het paar connectieven $\{\neg, \wedge\}$ is functioneel volledig.

Bewijs

Op een informele manier is dit reeds in hoofdstuk 2 'bewezen'. We herformuleren de bewering van de stelling als volgt:

Voor elke formule φ is er een logisch equivalente formule φ' die alleen met $\neg$ en $\wedge$ geconstrueerd is.

Basisstap

Voor atomaire formules geldt dit triviaal, want atomaire formules bevatten geen connectieven.

Inductiestap

De inductiehypothese zegt dat we twee formules φ en ψ herschreven hebben tot logisch equivalente formules φ' respectievelijk ψ' die alleen met $\neg$ en $\wedge$ geconstrueerd zijn. We moeten laten zien dat dan ook $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ en $\varphi \leftrightarrow \psi$ aldus te herschrijven zijn. Voor negatie en conjunctie is dit direct duidelijk: $(\neg\varphi)' = \neg\varphi'$ en $(\varphi \wedge \psi)' = \varphi' \wedge \psi'$. Voor de andere connectieven gebruiken we tevens:

1 $(\varphi \vee \psi)' = \neg(\neg\varphi' \wedge \neg\psi')$
2 $(\varphi \rightarrow \psi)' = \neg(\varphi' \wedge \neg\psi')$
3 $(\varphi \leftrightarrow \psi)' = (\varphi \rightarrow \psi)' \wedge (\psi \rightarrow \varphi)'$ □

5.4 DUALITEIT

In het voorgaande zijn er diverse parallellen geweest tussen wetten voor conjunctie en disjunctie. Zo ontstaat uit de ene wet van De Morgan $\neg(\varphi \vee \psi) \leftrightarrow (\neg\varphi \wedge \neg\psi)$ de andere, als we $\wedge$ en $\vee$ omwisselen: $\neg(\varphi \wedge \psi) \leftrightarrow (\neg\varphi \vee \neg\psi)$. Iets soortgelijks geldt voor de twee distributiviteitswetten. We kunnen dit verschijnsel als volgt zichtbaar maken. Laat φ een formule zijn waarin alleen de connectieven $\wedge$, $\vee$, $\neg$ voorkomen. De *duale* formule van φ (notatie: φ^d) is de formule die ontstaat door elk voorkomen van $\wedge$ in φ te vervangen door $\vee$ en elk voorkomen van $\vee$ door $\wedge$.

Voorbeeld 5.2

- $(p \vee q)^d = (p \wedge q)$
- $(\neg p \vee (p \wedge q))^d = (\neg p \wedge (p \vee q))$

STELLING 5.2

Dualiteit
Laat φ, ψ formules zijn waarin alleen de connectieven $\wedge, \vee, \neg$ voorkomen. Dan geldt:

$$\models \varphi \leftrightarrow \psi \Rightarrow \models \varphi^{d} \leftrightarrow \psi^{d}$$

Bewijs

Laat voor een willekeurige formule χ, χ^{+} de formule zijn die ontstaat door in χ^{d} alle atomen te vervangen door hun negaties. Met formule–inductie bewijzen we nu eerst het volgende:

BEWERING 5.2

$$\models \neg\chi \leftrightarrow \chi^{+}$$

Bewijs

Basisstap

Inductiestap

Als χ een atoom is dan is $\chi^{d} = \chi$ en dus $\chi^{+} = \neg\chi$.
Stel de bewering is juist voor α en β: $\models \neg\alpha \leftrightarrow \alpha^{+}$ en $\models \neg\beta \leftrightarrow \beta^{+}$.
Geval negatie: Merk op dat $(\neg\alpha)^{+} = \neg(\alpha^{+})$. Samen met de inductiehypothese geldt: $(\neg\alpha)^{+} = \neg(\alpha^{+}) \leftrightarrow \neg\neg\alpha$.
Geval conjunctie: Wegens de definitie van dualiteit geldt $(\alpha \wedge \beta)^{+} = (\alpha^{+} \vee \beta^{+})$. Volgens de inductiehypothese is $(\alpha^{+} \vee \beta^{+})$ equivalent met $(\neg\alpha \vee \neg\beta)$. Volgens de tweede wet van De Morgan is $(\neg\alpha \vee \neg\beta)$ equivalent met $\neg(\alpha \wedge \beta)$. Dus $\models (\alpha \wedge \beta)^{+} \leftrightarrow \neg(\alpha \wedge \beta)$.
Disjunctie gaat analoog. □

Merk eerst op dat als we de atomen van χ^{+} vervangen door hun negatie, er een formule $\chi^{+\prime}$ ontstaat die op grond van de equivalentie van dubbele negaties logisch equivalent is met χ^{d}.
Dan vervolgen we nu het bewijs van stelling 5.2.

Stel $\models \varphi \leftrightarrow \psi$. Dan ook $\models \neg\varphi \leftrightarrow \neg\psi$. Uit de zojuist bewezen bewering volgt dan de logische equivalentie $\models \varphi^{+} \leftrightarrow \psi^{+}$. Met behulp van substitutie volgt dan $\models \varphi^{+\prime} \leftrightarrow \psi^{+\prime}$ en dus $\models \varphi^{d} \leftrightarrow \psi^{d}$. □

5.5 ADEQUAATHEID VAN SEMANTISCHE TABLEAUS

In hoofdstuk 3 is de adequaatsheidsstelling geformuleerd. De adequaatheidsstelling zegt dat semantische tableaus precies in staat zijn om geldige gevolgtrekkingen te testen. We gaan deze stelling nu bewijzen. In het bewijs gebruiken we een bewering voor gesloten en een bewering voor open tableaus. De bewijzen van die beide beweringen volgen nu eerst.

Gesloten tableau

In de eerste bewering gebruiken we een uitbreiding van de notie semantisch geldig: voor rijtjes formules Φ en Ψ betekent $\Phi \models \Psi$ dat elke waardering die alle formules in Φ de waarde 1 geeft, minstens één formule in Ψ de waarde 1 geeft. Als het rijtje Ψ uit één formule ψ bestaat, reduceert dit tot semantische geldigheid in de gebruikelijke zin.

BEWERING 5.3

Stel $\Phi \circ \Psi$ is de topsequent van een gesloten tableau. Dan geldt $\Phi \models \Psi$.

Bewijs

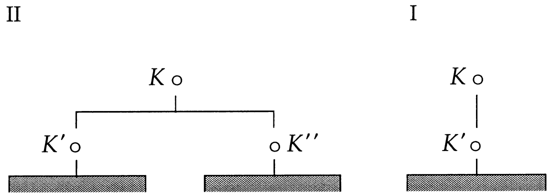
Hier is een gewone wiskundige inductie het meest aangewezen, en wel naar het aantal knopen in gesloten tableaus (dit zijn er altijd slechts eindig veel).

Basisstap

Het tableau heeft één knoop. Sluiting in de bovenste knoop betekent dat een zekere formule zowel links in Φ als rechts in Ψ voorkwam. Dan is $\Phi \models \Psi$ een geldige gevolgtrekking.

Inductiestap

Het tableau heeft meer knopen. Dan is het ontstaan door een van de regels voor connectieven toe te passen. Dit kan via een splitsende regel (type II) of een niet–splitsende regel (type I) zijn gebeurd:



De gearceerde gebieden staan voor subbomen. Dat zijn zelf ook gesloten tableaus, maar met minder knopen, zodat de inductiehypothese op hun topknoop van toepassing is. Aan de hand van twee gevallen zullen we illustreren hoe nu de gewenste bewering wordt bereikt.
Type I: regel $\neg_L$. De topsequent heeft dan de vorm: $\Phi, \neg\alpha \circ \Psi$; en de sequent onmiddellijk daaronder: $\Phi \circ \Psi, \alpha$. Volgens de inductiehypothese geldt: $\Phi \models \Psi, \alpha$. Zij nu V een willekeurige waardering die alle formules in Φ en $\neg\alpha$ waar maakt. Omdat V Φ waar maakt, volgt uit $\Phi \models \Psi, \alpha$ dat een formule uit Ψ of α waar wordt gemaakt. Het laatste strijdt met de aanname dat V $\neg\alpha$ waar maakt en dus maakt V een formule uit Ψ waar, dus $\Phi, \neg\alpha \models \Psi$.
Type II: regel $\wedge$R. De topsequent heeft nu de vorm $\Phi \circ \alpha \wedge \beta, \Psi$. De twee onmiddellijke opvolgers zijn: $\Phi \circ \alpha, \Psi$ en $\Phi \circ \beta, \Psi$. Met de inductiehypothese geldt $\Phi \models \alpha, \Psi$ en $\Phi \models \beta, \Psi$. Maar dan maakt iedere waardering die alle formules uit Φ waar maakt, hetzij een formule uit Ψ waar, hetzij zowel α als β: en dus ook $\alpha \wedge \beta$. Dus $\Phi \models \alpha \wedge \beta, \Psi$. □

Open tableau

Nu bekijken we het geval van een open tableau.

BEWERING 5.4

Een open tableau heeft een tak τ die niet sluit. Zij V de waardering die alle propositieletters die links op de tak τ voorkomen, waar maakt en alle propositieletters die rechts voorkomen onwaar. Dan geldt voor elke formule φ: als φ links op de tak voorkomt, dan $V(\varphi) = 1$, en als φ rechts op de tak voorkomt, dan $V(\varphi) = 0$.

Bewijs

Deze keer is een formule–inductie de passende bewijsvorm. We kunnen de bewering aantonen met inductie naar de opbouw van formules, gebruikmakend van het feit dat in een niet–sluitend semantisch tableau elke toepasbare regel ook werkelijk is toegepast.

Basisstap

Atomen. De bewering geldt dan per definitie voor de valuatie V.

Inductiestap

Stel dat de bewering reeds opgaat voor formules φ en ψ. Weer beschouwen we slechts een enkel illustratief geval.
Negatie links: Als $\neg\varphi$ links op de tak voorkomt, dan moet daaronder een keer de regel $\neg_L$ zijn toegepast, zodat φ rechts voorkomt. Volgens de inductiehypothese geldt dan $V(\varphi) = 0$, zodat $V(\neg\varphi) = 1$ volgens de waarheidstabel voor $\neg$.
Conjunctie rechts: Als $\varphi \wedge \psi$ rechts op τ voorkomt, dan moet daaronder een keer de regel $\wedge_R$ zijn toegepast, zodat een splitsing ontstond. De tak τ volgt hierin de linker– of rechterkant, bijvoorbeeld de linkerkant waarin we φ rechts in de sequent terugvinden. Volgens de inductie-hypothese geldt dan $V(\varphi) = 0$, zodat tevens $V(\varphi \wedge \psi) = 0$. □

Met de twee voorgaande beweringen 5.3 en 5.4 kunnen we nu de adequaatheidsstelling bewijzen:

STELLING 5.3

Adequaatheid
Als $\varphi_1, ..., \varphi_n$ en ψ formules zijn, dan geldt: $\varphi_1, ..., \varphi_n \models \psi$ desda er is een gesloten tableau voor $\varphi_1, ..., \varphi_n \circ \psi$.

Bewijs

$\Leftarrow$ Dit is een speciaal geval van bewering 5.3.
$\Rightarrow$ Stel er is geen gesloten tableau voor deze sequent. Omdat een tableau op zich altijd te construeren is, moet er dan een open tableau zijn. Op grond van bewering 5.4 levert elke open tak daarvan een tegenvoorbeeld, zodat geldigheid is weerlegd. □

Uit deze stelling kunnen we tevens concluderen dat bij de tableau-methode de manier waarop we de tableauconstructie aanpakken (de volgorde waarin we regels toepassen ligt immers niet vast) niet van invloed is op het verkregen antwoord:

GEVOLG 5.1

Als één tableau voor een sequent sluit, dan sluiten alle tableaus daarvoor.

Bewijs

Als een tableau sluit, dan is de corresponderende gevolgtrekking geldig. Als er tevens een open tableau zou zijn, dan was de corresponderende gevolgtrekking niet geldig. Tegenspraak. Dus alle tableaus sluiten. □

5.6 CORRECTHEID EN VOLLEDIGHEID

Ten slotte komen we terug op het verband tussen geldige gevolgtrekkingen en afleidbaarheid met natuurlijke deductie in de propositielogica, aangestipt in hoofdstuk 4. Dit wordt gegeven in de volledigheidsstelling:

STELLING 5.4

Volledigheid van de propositielogica
Als Σ een verzameling formules is en φ een formule, dan geldt:

$$\Sigma \vdash \varphi \Leftrightarrow \Sigma \models \varphi$$

Met een lege verzameling Σ zegt dit resultaat in het bijzonder dat de tautologieën van de propositielogica precies de stellingen van het bewijssysteem van natuurlijke deductie zijn.
We zullen een bewijs van deze stelling schetsen, ter wille van het gemak voor een taal zonder het equivalentiesymbool. (Merk op dat $\varphi \leftrightarrow \psi$ equivalent is met $(\varphi \to \psi) \wedge (\psi \to \varphi)$.) Zoals reeds eerder opgemerkt, valt de stelling uiteen in twee richtingen.

Correctheid

De eerste daarvan is semantische *correctheid* van het bewijssysteem:

STELLING 5.4a

$$\Sigma \vdash \varphi \Rightarrow \Sigma \models \varphi$$

Bewijs

Het bewijs gaat met inductie naar het aantal knopen in een afleidingsboom B voor φ uit Σ.

Basisstap

Een boom met één knoop is een aanname op zich. Dan geldt kennelijk: φ zit in Σ, en dus $\Sigma \models \varphi$.

Inductiestap

Een boom met meer knopen is ontstaan door toepassing van een laatste bewijsregel. We moeten nu systematisch nagaan dat alle bewijsregels van natuurlijke deductie 'geldigheid bewaren'. We geven twee kenmerkende illustraties:

1 Stel B eindigt met een toepassing van $\neg$E. Dan hebben we een formule ψ afgeleid uit $\Sigma = \Sigma_1 \cup \Sigma_2$ waar Σ_1 zekere φ afleidt met minder stappen en Σ_2 $\neg\varphi$, ook met minder stappen. Volgens de inductiehypothese geldt dan $\Sigma_1 \vDash \varphi$ en $\Sigma_2 \vDash \neg\varphi$. Maar dan kan er geen model zijn voor Σ (omdat dat zowel φ als $\neg\varphi$ waar zou moeten maken): dus geldt triviaal dat elk model van Σ model van ψ is, zodat $\Sigma \vDash \psi$.
2 Stel B eindigt met een toepassing van $\rightarrow$I. Dan heeft B conclusie $\varphi \rightarrow \psi$ afgeleid uit Σ. Als we de laatste knoop van B weghalen, dan blijft een boom B_1 over met één knoop minder, aannames Σ_1 en conclusie ψ (waarbij $\Sigma = \Sigma_1 \setminus \{\varphi\}$).
Zij V nu een willekeurige waardering die model is van Σ. Aan te tonen is dat ook $V(\varphi \rightarrow \psi) = 1$. Daartoe volstaat, gezien de waarheidstabel voor implicatie, om te laten zien dat, als $V(\varphi) = 1$, dan ook $V(\psi) = 1$. Maar indien $V(\varphi) = 1$, dan geldt dat V model is van Σ_1, zodat de inductiehypothese op B_1 ons vertelt dat inderdaad ook $V(\psi) = 1$.
De andere gevallen gaan evenzo. □

Volledigheid

De tweede richting van deze stelling drukt de *volledigheid* uit van de propositielogica:

STELLING 5.4b

$$\Sigma \vDash \varphi \Rightarrow \Sigma \vdash \varphi$$

Het bewijzen hiervan heeft meer voeten in de aarde. Om te beginnen voeren we nog een notie rond het begrip consistentie in. Het is duidelijk dat elke deelverzameling van een consistente verzameling zelf weer consistent is. Voor uitbreiding van consistente verzamelingen is dat soms wel en soms niet het geval.

Voorbeeld 5.3

Als aan de consistente verzameling $\{p \vee \neg q, p \rightarrow \neg q\}$ de formule q wordt toegevoegd, dan is de nieuwe verzameling niet meer consistent. Als we daarentegen $\{p \vee \neg q, p \rightarrow \neg q\}$ uitbreiden met $\neg q$, dan is de uitgebreide verzameling wel consistent, en hetzelfde geldt als we r zouden toevoegen. Hoever kunnen we een dergelijk uitbreidingsproces eigenlijk voortzetten?

Maximaal consistent

Als een consistente verzameling Γ niet meer uit te breiden is tot een grotere consistente verzameling, dan heet Γ maximaal consistent.

DEFINITIE 5.1

Maximaal consistent
Een formuleverzameling Γ heet *maximaal consistent* als geldt:
a Γ is consistent;
b iedere echte uitbreiding van Γ is inconsistent.

Er valt te bewijzen dat elke consistente verzameling uit te breiden is tot een maximaal consistente verzameling.

STELLING 5.5

'Lemma van Lindenbaum'
Elke consistente verzameling Γ is bevat in een maximaal consistente verzameling Γ^*.

Bewijs

Ter wille van de overzichtelijkheid geven we een bewijs voor een taal met een aftelbaar alfabet. In dat geval is een formule een eindige reeks symbolen uit een aftelbaar alfabet en zijn er dus aftelbaar veel formules in de propositielogica, dat wil zeggen, er bestaat een opsomming φ_0, φ_1, φ_2, ... van alle formules van de taal. We gaan nu een gegeven consistente verzameling Γ systematisch stapsgewijs uitbreiden tot een grotere consistente verzameling:

$$\Gamma_0 = \Gamma$$

$$\begin{array}{lll} \Gamma_{i+1} = & \Gamma_i \cup \{\varphi_i\} & \text{als } \Gamma_i \cup \{\varphi_i\} \text{ consistent is} \\ \Gamma_{i+1} = & \Gamma_i & \text{anders} \end{array}$$

In de limiet ontstaat dan een verzameling Γ^* als volgt:

$$\Gamma^* = \bigcup_i \Gamma_i$$

Iedere Γ_i is consistent en Γ^* is maximaal consistent:
1 Elke Γ_i is consistent.
Dit bewijzen we met natuurlijke inductie naar i. Basisstap: $\Gamma_0 = \Gamma$, dus Γ_0 is consistent. Inductiestap: stel Γ_i is consistent. Als $\Gamma_i \cup \{\varphi_i\}$ consistent is, dan is $\Gamma_{i+1} = \Gamma_i \cup \{\varphi_i\}$ ook consistent, anders blijft Γ_{i+1} de reeds consistente Γ_i.
2 Dan is ook Γ^* consistent.
Stel van niet, dan is er een formule ξ met zowel $\Gamma^* \vdash \xi$ als $\Gamma^* \vdash \neg\xi$. Omdat elke afleiding slechts *eindig* veel premissen gebruikt, is er een voldoende groot getal k te vinden zodat alle premissen die in de afleidingen voor ξ en $\neg\xi$ gebruikt worden, reeds in Γ_k zitten, ofwel $\Gamma_k \vdash \xi$ en $\Gamma_k \vdash \neg\xi$. Maar dan is Γ_k inconsistent, in tegenspraak met het feit dat elke Γ_i consistent was.
3 Γ^* is maximaal.
We nemen een formule $\psi \notin \Gamma^*$. In de aftelling van alle propositionele formules is er een k waarvoor $\psi = \varphi_k$. Formule ψ is kennelijk niet toegevoegd aan Γ_k om Γ_{k+1} te maken, want dan zou ψ ook in Γ^* zitten.

Dit betekent dat $\Gamma_k \cup \{\psi\}$ inconsistent is. Maar dan is $\Gamma^* \cup \{\psi\}$ ook inconsistent, want Γ_k is een deel van Γ^*. □

Maximaal consistente verzamelingen hebben enkele plezierige eigenschappen.

BEWERING 5.5

Als Γ een maximaal consistente verzameling is, dan geldt voor elke formule φ: $\varphi \in \Gamma \Leftrightarrow \Gamma \vdash \varphi$.

Bewijs

$\Rightarrow$ Dit is duidelijk.
$\Leftarrow$ Stel $\varphi \notin \Gamma$. Omdat Γ maximaal consistent is, is $\Gamma \cup \{\varphi\}$ inconsistent. Hieruit volgt dat $\Gamma \vdash \neg\varphi$ (bewering 4.2). Als nu ook $\Gamma \vdash \varphi$, dan zou Γ inconsistent zijn. Tegenspraak, dus $\Gamma \nvdash \varphi$. □

Vervolgens kunnen we constateren dat maximaal consistente verzamelingen in zekere zin formules 'afbreken':

BEWERING 5.6

Decompositie
Als Γ een maximaal consistente verzameling is, dan geldt:
a $\neg\varphi \in \Gamma \Leftrightarrow \varphi \notin \Gamma$
b $\varphi \wedge \psi \in \Gamma \Leftrightarrow \varphi \in \Gamma$ en $\psi \in \Gamma$
c $\varphi \vee \psi \in \Gamma \Leftrightarrow \varphi \in \Gamma$ of $\psi \in \Gamma$
d $\varphi \rightarrow \psi \in \Gamma \Leftrightarrow$ als $\varphi \in \Gamma$, dan $\psi \in \Gamma$

Bewijs

We bewijzen twee kenmerkende gevallen.
Geval a
$\Rightarrow$ Stel $\neg\varphi \in \Gamma$. Dan zeker $\Gamma \vdash \neg\varphi$. Omdat Γ consistent is, geldt dan $\Gamma \nvdash \varphi$, zodat met bewering 5.5 volgt: $\varphi \notin \Gamma$.
$\Leftarrow$ Stel $\neg\varphi \notin \Gamma$. Dan is, wegens de maximale consistentie van Γ, $\Gamma \cup \{\neg\varphi\}$ inconsistent. Volgens bewering 4.2 geldt dan: $\Gamma \vdash \varphi$, zodat weer $\varphi \in \Gamma$.

Geval b
$\Rightarrow$ Stel $\varphi \wedge \psi \in \Gamma$. Dan geldt zeker $\Gamma \vdash \varphi \wedge \psi$. De eliminatieregels voor $\wedge$ geven nu: $\Gamma \vdash \varphi$ en $\Gamma \vdash \psi$. Dit impliceert volgens bewering 5.5 weer $\varphi \in \Gamma$ en $\psi \in \Gamma$.
$\Leftarrow$ Stel dat $\varphi \in \Gamma$ en $\psi \in \Gamma$. Dus: $\Gamma \vdash \varphi$ en $\Gamma \vdash \psi$ en de introductieregel voor $\wedge$ leveren dan: $\Gamma \vdash \varphi \wedge \psi$, zodat dan volgens bewering 5.5 weer geldt: $\varphi \wedge \psi \in \Gamma$. □

Met deze resultaten kunnen we verder gaan met het bewijs van de volledigheid van de propositielogica. Hoe kunnen we nu uit het tamelijk abstracte gegeven van semantische geldigheid concluderen tot het bestaan van een zo concreet syntactisch object als een natuurlijke deductieboom? Hiertoe gebruiken we weer contrapositie. Op die manier kunnen we een equivalente implicatie bewijzen, die toch van een syntactisch gegeven naar een semantische conclusie voert:

STELLING 5.4b′

$\Sigma \nvdash \psi \Rightarrow \Sigma \nvDash \psi$

Het bewijs loopt nu als volgt: Bewering 4.2 zegt dat uit $\Sigma \nvdash \psi$ volgt dat de verzameling formules $\Sigma \cup \{\neg\psi\}$ *syntactisch consistent* is. We zoeken een waardering V die alle formules in deze verzameling waar maakt. Zo'n waardering levert een semantisch tegenvoorbeeld dat aantoont dat $\Sigma \nvDash \psi$.

Het volstaat dus om te bewijzen dat we voor elke syntactisch consistente verzameling formules Γ een model kunnen vinden. Uit 'Lindenbaums Lemma' (stelling 5.5) volgt dat Γ is uit te breiden tot een *maximaal consistente* verzameling Γ^*. Met behulp van bewering 5.6 over 'decompositie' construeren we een model voor Γ^*:

BEWERING 5.7

Voor elke maximaal consistente verzameling formules Γ^* bestaat een waardering V zodat voor elke formule φ geldt: $\varphi \in \Gamma^* \Leftrightarrow V(\varphi) = 1$.

Bewijs

Definieer V als volgt: $V(p) = 1$ voor iedere atomaire p uit Γ^*.
Vanzelfsprekend geldt nu: $p \in \Gamma^* \Leftrightarrow V(p) = 1$.
Deze equivalentie geldt tevens voor meer complexe formules ξ. Dit bewijzen we met formule–inductie:

Basisstap

Voor atomaire formules ξ geldt de bewering reeds wegens de definitie van V.

Inductiestap

Stel dat de equivalentie reeds geldt voor φ en ψ.
Geval $\neg\varphi$: $V(\neg\varphi) = 1 \Leftrightarrow$ (waarheidstabel) $V(\varphi) \neq 1 \Leftrightarrow$ (inductiehypothese) $\varphi \notin \Gamma^* \Leftrightarrow$ (bewering 5.6) $\neg\varphi \in \Gamma^*$.
Geval $\varphi \rightarrow \psi$: $V(\varphi \rightarrow \psi) = 1 \Leftrightarrow$ (waarheidstabel) $V(\varphi) \neq 1$ of $V(\psi) = 1 \Leftrightarrow$ (inductiehypothese) $\varphi \notin \Gamma^*$ of $\psi \in \Gamma^* \Leftrightarrow$ (bewering 5.6) $\varphi \rightarrow \psi \in \Gamma^*$.
Hiermee is de inductiestap voltooid, en daarmee tevens het bewijs van de bewering. □

Met bewering 5.7 bewijzen we stelling 5.4b′ en de daarmee equivalente stelling 5.4b, en hiermee is het bewijs van de volledigheidsstelling 5.4 voltooid.

5.7 BESLISBAARHEID

De vorige resultaten gaven aan dat onze bewijssystemen in zekere zin semantisch adequaat zijn. Een heel ander soort metavraag betreft de algoritmische *complexiteit* van die bewijssystemen. Met name de vraag of het mogelijk is om, gegeven een of andere formuleverzameling Σ, door middel van een *effectieve* mechanische procedure uit te maken of een gegeven formule φ uit Σ volgt. Met zo'n procedure bedoelen we een echt algoritme dat deze vraag in een eindig aantal stappen kan beslissen en dat in principe programmeerbaar zou zijn op een computer.
Zo'n probleem wordt een *beslisbaarheidsprobleem* genoemd. Bijvoorbeeld in het geval van natuurlijke deductie hebben we al in hoofdstuk 4 beargumenteerd dat de afleidbaarheidsvraag althans a priori open ligt: de zoekruimte van mogelijke afleidingen voor een formule is in principe oneindig. Maar ook bij andere logische noties kan men beslisbaarheidsvragen stellen: bijvoorbeeld, is het beslisbaar of een gegeven stel connectieven functioneel volledig is?
In het algemeen blijken beslisbaarheidsvragen voor de propositielogica een positief antwoord te hebben. Zo levert de volledigheidsstelling ons reeds een dergelijk antwoord op in het geval van natuurlijke deductie: afleidbaarheid is immers equivalent met semantisch geldig gevolg en deze laatste notie is beslisbaar (met waarheidstabellen of tableaus).

Precieze formuleringen van beslissingsalgoritmen voor de logica vallen buiten het bestek van dit boek. Een viertal benaderingen van het begrip berekenbaar worden in hoofdstuk 14 uiteengezet. Zie ook hoofdstuk 17.

5.8 OPGAVEN

5.1 Een connectief c_1 *regeert* een connectief c_2 als c_2 voorkomt in het bereik van c_1. De *connectiefdiepte* van een formule is de maximale lengte van een rij ('nest') van connectieven die elkaar regeren. Bijvoorbeeld voor $((\neg p \wedge q) \vee r)$ is dit 3: $\vee$ regeert $\wedge$ en $\wedge$ regeert $\neg$.
Definieer de connectiefdiepte met inductie op formules.

5.2 Een bijzondere vorm van inductie is *simultane inductie*. Hierbij worden meerdere begrippen tegelijk inductief gedefinieerd zodanig dat in de inductieve definitie van elk van de begrippen de andere begrippen voorkomen.
Welke begrippen A en B worden gedefinieerd met behulp van de volgende inductie (denk aan formules als bomen)?

$$\begin{array}{ll} A(p) & = 1 \\ A(\neg\psi) & = A(\psi) + 1 \\ A(\psi \wedge \chi) & = max(A(\psi), A(\chi)) + 1 \end{array}$$

$$\begin{array}{ll} B(p) & = 1 \\ B(\neg\psi) & = max(B(\psi), A(\psi) + 1) \end{array}$$

$$B(\psi \wedge \chi) = max(B(\psi), B(\chi), A(\psi) + A(\chi) + 1)$$

5.3 Bewijs met inductie dat in geen formule φ een paar aangrenzende haakjes ')(' of ' () ' voorkomt.

5.4 Bewijs met inductie dat
i elk echt beginstuk van een formule óf alleen uit symbolen $\neg$ bestaat óf meer linker– dan rechterhaakjes bevat.
ii elke formule slechts één constructieboom heeft. (Dit heet *unieke leesbaarheid*.)

5.5 Bewijs dat de connectieven $\wedge$, $\vee$ niet functioneel volledig zijn door een geschikte bewering te formuleren en te bewijzen over de waarden van al zulke formules onder de waardering met louter enen.

5.6 De *Poolse notatie* voor formules is gedefinieerd met behulp van de volgende inductieve opzet:
- atomen zijn formules;
- als φ, ψ formules zijn, dan ook $\neg\varphi$, $\wedge\ \varphi\ \psi$, $\vee\ \varphi\ \psi$, $\rightarrow\ \varphi\ \psi$ en $\leftrightarrow\ \varphi\ \psi$.

Voorbeelden:
- $(p \rightarrow \neg q)$ vertaald in het Pools: $\rightarrow p\ \neg q$.
- Wél Pools: $\rightarrow \wedge\ p\ q \wedge \neg r\ q$; niet Pools: $\rightarrow \wedge \wedge\ p\ q\ r$.

De Poolse notatie is de wiskundige standaardnotatie voor functies.
a Vertaal van 'gewoon' naar Pools: $(\neg(p \rightarrow (\neg q \leftrightarrow r)) \vee \neg r)$.
b En omgekeerd: $\wedge \wedge p \neg \vee q\ r\ r$.

c Bewijs dat het volgende algoritme alle Poolse formules herkent:
'Ken als volgt getallen toe:

atomen $\mapsto -1$
$\neg$ $\mapsto 0$
$\wedge, \vee, \rightarrow, \leftrightarrow \mapsto +1$

Om te bepalen of een string een formule is: schrijf onder de symbolen hun kengetallen en tel die op, van links naar rechts.
Als de eindsom −1 is en alle tussensommen ≥ 0 zijn, dan is er sprake van een Poolse formule.'
(Maak enkele voorbeelden.)

* d Bewijs dat dit algoritme *alleen* Poolse formules herkent.

* 5.7 Voor twee waarderingen V_1 en V_2 definiëren we:

$V_1 \leq_p V_2$ indien $V_1(p) \leq V_2(p)$ en $V_1(q) = V_2(q)$ voor alle $q \neq p$

Een formule φ heet *monotoon in p* als geldt $V_1 \leq_p V_2 \Rightarrow V_1(\varphi) \leq V_2(\varphi)$.
Tracht te bepalen onder welke voorwaarden op de voorkomens van p in φ van monotonie sprake zal zijn.

* 5.8 Laat zien dat de consistentiestelling, die zegt dat elke consistente verzameling formules een model heeft, *equivalent* is met de volledigheidsstelling.

Blok III

Predikaatlogica

Inhoud hoofdstuk 6

Predikaatlogica: taal

Hoofdstuk 6

Predikaatlogica: taal

6.1 INLEIDING

Beschouw de volgende gevolgtrekking:

aanname	Alle mannen voetballen
aanname	Alle fietsen zijn blauw
conclusie	Alle mannen voetballen en alle fietsen zijn blauw

Propositioneel kan dit worden weergegeven door:

$p, q \,/\, p \wedge q$

Dit is in de propositielogica een geldige gevolgtrekking.
Een andere, minstens even eenvoudige redenering lijkt de volgende:

aanname	Alle mannen voetballen
aanname	Piet is een man
conclusie	Piet voetbalt

Hoe zouden we dit in de propositielogica kunnen weergeven? 'Alle mannen voetballen' kan slechts met een atomaire formule worden weergegeven, evenals de andere twee zinnen. Maar dan ziet deze gevolgtrekking er als volgt uit:

$p, q \,/\, r$

En dit is geen geldige redenering in de propositielogica. Het probleem is dat een zin meer informatie kan bevatten die voor de gevolgtrekking belangrijk is, dan we propositioneel weer kunnen geven: bijvoorbeeld bij het beschrijven van eigenschappen ('predikaten') van individuen. Het wordt dan kennelijk nodig om de taal van de propositielogica uit te breiden, zodat meer *interne structuur* van een zin blootgelegd kan worden.

De belangrijkste uitbreiding van de propositielogica heet de *predikaatlogica*. De predikaatlogica heeft naast propositionele connectieven ook middelen om predikaten weer te geven. Bovendien geeft zij de mogelijkheid hierop logische bewerkingen uit te voeren, zoals met name *kwantificatie*, uitgedrukt door een woord als 'alle'. Dit nieuwe systeem vormt het onderwerp van de volgende groep hoofdstukken.

6.2 TAALVERWERVING INFORMEEL

Net als de propositielogica heeft ook de predikaatlogica een eigen taal. Deze bevat net als de propositionele taal zowel logische als niet–logische symbolen. De taal van de predikaatlogica heeft echter meer uitdrukkingskracht dan die van de propositielogica, en is daardoor moeilijker te leren. We zullen daarom in dit hoofdstuk een aanpak kiezen die ook bij natuurlijke talen of programmeertalen niet onbekend is. Het gaat om twee zaken: u moet met de predikaatlogische taal praktisch leren 'lezen en schrijven' en u moet ook iets van de formele structuur van die taal begrijpen. Om vertrouwd te raken met wat in de taal gezegd kan worden, volgen we eerst de 'natuurmethode' en bespreken informeel een aantal typerende constructies. Daarna bekijken we de syntactische definitie van de taal meer precies, en ten slotte geven we al een aantal grammaticale verschijnselen die op grond van de formele syntax optreden: de predikaatlogische taal bevat namelijk enkele fundamentele noties, zoals 'bereik' of 'variabelenbinding', die ook van belang zijn in natuurlijke talen en programmeertalen.

Predikaten

De meest eenvoudige beweringen die we beschouwen, zijn predikaatzinnen zoals:

a Piet voetbalt.
b Jan bemint Marie.
c Steven is kind van Roel en Adriana.

We zien hier telkens een of meer individuen waarover iets 'geprediceerd' ('gezegd') wordt: een eigenschap in a, een binaire relatie in b, een ternaire relatie in c. Predikaatzinnen met meer dan drie argumenten zijn ook denkbaar. Ditzelfde basispatroon doet zich net zo voor binnen de wiskunde en informatica:

d 3 is een priemgetal.
e 3 is kleiner dan 5.
f Punt x ligt tussen y en z.

De taal van de predikaatlogica heeft symbolen voor predikaten. Deze noemen we ook wel predikaatletters. Iedere predikaatletter heeft een vast aantal argumenten. Dit noemen we de *plaatsigheid* van het predikaat. Tevens bevat de taal van de predikaatlogica eigennamen voor individuen, de zogenaamde *individuele constanten*. Predikaatletters noteren we met hoofdletters, individuele constanten met kleine letters. Bijvoorbeeld:

a′ $V(p)$
b′ $B(j, m)$
c′ $K(s, r, a)$

Er is hier ruimte voor verschillende conventies. Een deel van de informaticaliteratuur verkiest 'syntactic sugaring' tot notaties die op natuurlijke taal lijken, zoals:

a′′ VOETBALT(PIET)
b′′ BEMINT(JAN, MARIE)
c′′ KIND_VAN(STEVEN, ROEL, ADRIANA)

Sommige dialecten laten de haakjes weg, wat zonder verlies van eenduidigheid kan. Over het algemeen zullen wij deze conventie voortaan volgen. We krijgen dan:

a′′′ Vp
b′′′ Bjm
c′′′ $Ksra$

Alleen bij de formele invoering van begrippen grijpen we meestal terug op de notatie mét haakjes.

Merk op dat niet alle argumenten in een predikaatzin verschillend hoeven te zijn. Zo drukt *Bmm* uit dat Marie zichzelf bemint. Merk ook nog op dat we afspraken hanteren over de volgorde van argumenten: *Bjm* betekent dat Jan Marie bemint, *Bmj* daarentegen dat Marie Jan bemint.

Functies

Behalve predikaten komen we in de wetenschappelijke en computationele praktijk ook veel voorbeelden tegen van basisbeweringen die een ander aspect introduceren. Bijvoorbeeld, een algebraïsche vergelijking als

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

is een binaire predikaatzin van gelijkheid tussen twee objecten die op verschillende samengestelde manieren worden omschreven. Naast relaties en objecten willen we daarom ook *functies* in de taal weergeven. Deze komen overigens ook in de natuurlijke taal wel voor, getuige uitdrukkingen als 'Jans moeder', 'Roels leeftijd'. Functies worden in de predikaatlogica aangegeven met kleine letters, weer met een afspraak over hun plaatsigheid.

Complexe uitdrukkingen

Complexe uitdrukkingen ontstaan nu door combinatie van basispatronen.
De eerste manier waarop dit kan, is die van de propositielogica: we nemen gewoon alle eerdere connectieven over. Aldus ontstaan combinaties als:

Voorbeeld 6.1

a*	$Vp \rightarrow \neg Vr$	Als Piet voetbalt, dan voetbalt Roel niet.
b*	$Bjm \wedge Bmj$	Jan bemint Marie en Marie bemint Jan. ('Jan en Marie beminnen elkaar')

Kwantoren

Andere complexe uitdrukkingen zijn 'uitdrukkingen van hoeveelheid'. Die zeggen iets over hoeveel individuen al dan niet aan zekere predikaten voldoen. Dergelijke uitdrukkingen kunnen we logisch weergeven met behulp van de zogenaamde *kwantoren* $\forall$ en $\exists$.
$\forall$ wordt de *universele* kwantor genoemd en is te lezen als 'voor alle'.
$\exists$ wordt de de *existentiële* kwantor genoemd en is te lezen als 'er is een'.
Met $\forall$ respectievelijk $\exists$ kunnen zinnen van de volgende vormen worden weergegeven:
- '*Voor alle* objecten geldt ... '
- '*Er is minstens één* object waarvoor geldt ... '

Voorbeeld 6.2

Voorbeelden van zinnen met kwantoren in de natuurlijke taal en de wiskunde:
a Alle mannen voetballen.
b Fietsen kunnen blauw zijn.
c Geen getal is groter dan 4.
d Voor elke lijn l geldt: er is een lijn l' zodat l en l' parallel zijn.
(Merk op dat zinnen niet waar hoeven te zijn.)

Individuele variabelen

Het gebruik van kwantoren introduceert nog een nieuwigheid. Om de individuen aan te geven waarover gekwantificeerd wordt, markeren we de relevante posities in de betreffende zin met zogenaamde *individuele variabelen*, die we noteren met kleine letters x, y, z, ... Als we de vorige voorbeelden op deze manier herschrijven, krijgen we:

Voorbeeld 6.3

a′ $\forall x\,(Mx \rightarrow Vx)$ ('Ieder individu dat een man is, voetbalt')
b′ $\exists x\,(Fx \wedge Bx)$ ('Er is een individu dat een fiets is en blauw is')
c′ $\neg\exists x\,(Gx \wedge x > 4)$
d′ $\forall x\,(Lx \rightarrow \exists y\,(Ly \wedge Pxy))$

Meerdere kwantoren in één zin

De kracht van de predikaatlogische taal schuilt in de vele complexe combinaties die te maken zijn met propositionele connectieven en kwantoren. Bijvoorbeeld, veel voorkomende kwantorcombinaties zijn:

$\forall\exists$ Bij elk getal is er een groter getal.
Iedereen heeft een moeder.
$\exists\forall$ Er is een verzameling die bevat is in elke verzameling.
Iemand ('Eva') is voorouder van iedereen.

Zelfs begrijpelijke drievoudige combinaties komen voor, zoals in de bekende definitie van continuïteit van reële functies:

$\forall\exists\forall$ 'de functie f: $\mathbb{R} \rightarrow \mathbb{R}$ is continu in x'
$\forall \varepsilon > 0\ \exists \delta > 0\ \forall y \in \mathbb{R}\ (\,|x - y\,| < \delta \rightarrow |f(x) - f(y)\,| < \varepsilon)$

Ook in natuurlijke taal komen drievoudige combinaties voor:

'Iedereen haat soms wel eens iedereen.'

Nog langere combinaties zijn zeldzaam, en schijnen ons onmiddellijk bevattingsvermogen te buiten te gaan.

Alfabet

We definiëren nu formeel het alfabet van een predikaatlogische taal.

DEFINITIE 6.1

Alfabet
Het alfabet van een predikaatlogische taal L bestaat uit:
a een verzameling $\boldsymbol{C}$ van individuele constanten;
b een verzameling $\boldsymbol{P}$ van predikaatletters;
c een verzameling $\boldsymbol{F}$ van functieletters;
d logische symbolen $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$ en $\exists$;
e individuele variabelen;
f hulpsymbolen) en (.

Notatie

Voor individuele constanten gebruiken we meestal kleine letters a, b, c, ..., of ook wel a_1, a_2, a_3, ...
Voor predikaatletters nemen we in het algemeen hoofdletters P, Q, R, ..., of ook wel P_1, P_2, P_3, ... Desgewenst wordt ook de plaatsigheid van een

predikaatletter aangegeven door een hooggeplaatste index. Bijvoorbeeld P^1, P^2, P^3, ..., voor een–, twee–, dan wel drieplaatsige relaties, of $P^n{}_1$, $P^n{}_2$, $P^n{}_3$, ..., om meerdere relaties van een bepaalde plaatsigheid n aan te geven.
Voor functieletters worden meestal f, g, h, ... genomen, of ook wel f_1, f_2, f_3, ... Ook hier geven we de plaatsigheid aan door een hooggeplaatste index.
Voor individuele variabelen gebruiken we meestal de kleine letters uit het achterste deel van het alfabet u, v, w, x, y, z of ook wel x_1, x_2, x_3, ...
De logische symbolen, individuele variabelen en hulpsymbolen zijn de vaste onderdelen van elke predikaatlogische taal. De verzamelingen $\boldsymbol{C}$, $\boldsymbol{P}$ en/of $\boldsymbol{F}$ mogen ook leeg zijn.
Eigenlijk zouden we telkens moeten aangeven hoe de taal waarmee we werken, eruitziet. Meestal is dit echter wel duidelijk uit de context.
In het algemeen spreken we gewoon over 'de predikaatlogica' in plaats van 'een predikaatlogische taal' en passen de notatie aan al naar gelang de leesbaarheid en de elegantie dat vereisen.

6.3 SYNTAXIS VAN DE PREDIKAATLOGICA

Termen

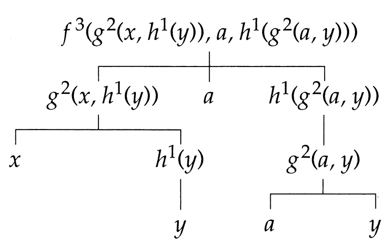
Een eerste belangrijke soort uitdrukkingen in onze taal zijn de zogenaamde *termen*. Termen duiden individuele objecten aan. Hier volgt een inductieve definitie van termen:

DEFINITIE 6.2

Termen
De termen van de predikaatlogica worden als volgt geconstrueerd:
a individuele variabelen en constanten zijn termen;
b als f een k–plaatsige functieletter is en t_1, ..., t_k zijn termen, dan is $f(t_1, ..., t_k)$ ook een term;
c niets is een term dan op grond van a en b.

Voorbeeld 6.4

$f^3(g^2(x, h^1(y)), a, h^1(g^2(a, y)))$ is een term met deze constructieboom:



Prefix– versus infixnotatie

Wat we hier gebruiken is de zogenaamde prefixnotatie, waarbij het functiesymbool vóór de argumenten komt. Bij vele tweeplaatsige wiskundige functies is echter de zogenaamde infixnotatie gebruikelijk, waarbij het functiesymbool tussen de argumenten komt. Dit is het geval bij optellen en vermenigvuldigen: $x + y$ en $x \cdot y$. De eerdere rekenregel $x \cdot (y + z) = x \cdot y + x \cdot z$ wordt in prefixnotatie als volgt geschreven: $\cdot\ (x, +(y, z)) = +(\cdot\ (x, y), \cdot\ (x, z))$. In de praktijk passen we ons bij ingeburgerde schrijfwijzen zoals in de wiskunde aan. Voor de algemene notatie blijft de prefixnotatie echter verkieslijk (evenals overigens binnen de wiskunde zelf het geval is). Eenzelfde opmerking geldt voor de later te bespreken predikaatletters, waar bijvoorbeeld ingeburgerde notaties voor bekende binaire relaties infix zijn, getuige $x < y$ ('kleiner dan') en $x \in y$ ('element van') en $x = y$ (gelijkheidspredikaat). Zodoende zouden we $x \cdot (y + z) = x \cdot y + x \cdot z$ officieel nog harder kunnen aanpakken, en wel als volgt: $=(\cdot\ (x, +(y, z)), +(\cdot\ (x, y), \cdot\ (x, z)))$.

Formules

Onze belangrijkste klasse van uitdrukkingen wordt gevormd door de eerdere beweringen ofwel *formules*. Deze worden weer inductief opgebouwd:

DEFINITIE 6.3

Formules

De formules van de predikaatlogica worden als volgt gedefinieerd:

a als P een k-plaatsige predikaatletter is en $t_1, ..., t_k$ zijn termen, dan is $P(t_1, ..., t_k)$ een formule;

b als φ en ψ formules zijn, dan ook $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ en $(\varphi \leftrightarrow \psi)$;

c als φ een formule is en x een individuele variabele, dan zijn $\forall x\ \varphi$ en $\exists x\ \varphi$ ook formules;

d niets is een formule dan op grond van a, b en c.

Een formule van de vorm $P(t_1, ..., t_k)$ heet wel een *atomaire* formule of een *atoom*. We geven een paar voorbeelden van formules.

Voorbeeld 6.5

- $P^2(x, a)$, $P^2(f^2(x, y), g^1(a))$ zijn atomaire formules.
- $\forall x\ (A^1(x) \rightarrow \forall y\ (R^2(x, y) \rightarrow A^1(y)))$, $\forall x \exists y\ \forall z\ (R^2(z, y) \leftrightarrow \neg Q^2(y, x))$ zijn formules.
- $P(\neg x)$, $A(x)\forall x$, $\forall x \exists R(x, y)$ zijn geen formules.

We herhalen nogmaals dat we in het vervolg soepel met notatie zullen omgaan, met zo weinig mogelijk boven– en onder–indices, zo weinig mogelijk interne en buitenste haakjes, en zoveel mogelijk gangbare schrijfwijzen.

Voorbeeld 6.6

We geven nu nog iets complexere voorbeelden van predikaatlogische formules. Om te beginnen uit de natuurlijke taal:

- $\neg\exists x\ \forall y\ Kxy$ — Niemand kent iedereen.
- $\forall x\ (\exists y\ Kxy \rightarrow \forall z\ Kxz)$ — Wie iemand kent, kent iedereen.
- $\forall x\ (\forall y\ (Kxy \leftrightarrow x = y) \rightarrow \neg\exists z\ Kzx)$ — Wie alleen zichzelf kent, wordt door niemand gekend.

Predikaatlogica en natuurlijke taal

Een voordeel van deze notatie ten opzichte van natuurlijke taal is, net als eerder bij de propositielogica, het ondubbelzinnige karakter. Bijvoorbeeld, 'elke man bemint een vrouw' heeft zowel een $\forall\exists$–lezing ('zijn moeder') als een daarmee niet equivalente $\exists\forall$–lezing ('Marilyn Monroe'). In predikaatlogische formules daarentegen liggen de kwantorvolgordes vast.

Predikaatlogica en wiskunde

De predikaatlogica is echter ook geschikt om wiskundige beweringen te formaliseren, bijvoorbeeld in de rekenkunde, meetkunde, enzovoort. De hele verzamelingenleer kan in predikaatlogica worden opgeschreven en daarmee codeert deze taal vrijwel alle bestaande wiskunde!

Voorbeeld 6.7

- uit de rekenkunde

'Bij elk natuurlijk getal is er een onmiddellijke opvolger':
$\forall x\ (\mathbb{N}x \rightarrow \exists y\ (\mathbb{N}y \wedge x < y \wedge \neg\exists z\ (\mathbb{N}z \wedge x < z \wedge z < y)))$

- uit de meetkunde

'Axioma van Pasch: een lijn vanuit een hoekpunt van een driehoek, getrokken door het inwendige, snijdt de overstaande zijde'. Dit wordt uitgedrukt in de volgende bewering over punten in het platte vlak met de ternaire relatie $Txyz$ gedefinieerd als 'y ligt tussen x en z' (dat wil zeggen: x, y en z liggen 'achter elkaar' op één lijn):
$\forall x\ \forall y\ \forall z\ \forall u\ \forall v((Tyxv \wedge Tzvu) \rightarrow \exists w\ (Tywz \wedge Tuxw))$

- uit de analyse

'De functie f: $\mathbb{R} \rightarrow \mathbb{R}$ is continu in x':
$\forall \varepsilon > 0\ \exists \delta > 0\ \forall y \in \mathbb{R}\ (\,|x - y\,| < \delta \rightarrow |f(x) - f(y)\,| < \varepsilon)$

- uit de verzamelingenleer

'Principe van extensionaliteit' (als twee verzamelingen dezelfde elementen hebben dan zijn ze gelijk): $\forall x\ \forall y\ (\forall z\ (z \in x \leftrightarrow z \in y) \rightarrow x = y)$.

Ondanks deze grote uitdrukkingskracht zijn er nog wel uitbreidingen van de predikaatlogica denkbaar. Zo bevat de categorie der kwantificerende uitdrukkingen in de natuurlijke taal nog vele andere begrippen, zoals 'minstens 3', 'precies 10' maar ook 'de meeste', 'veel'. Sommige

hiervan zijn binnen de predikaatlogica definieerbaar, andere niet. Zie hoofdstuk 12 voor een nader overzicht.
De predikaatlogica kan ook gebruikt worden als programmeertaal. De programmeertaal PROLOG (PROgrammation en LOGique), nader te bespreken in hoofdstuk 16, is een voorbeeld van een taal die gebaseerd is op predikaatlogica.

6.4 GRAMMATICALE VERSCHIJNSELEN

Predikaatlogische formules illustreren veel verschijnselen die bij het gebruik van operatoren (functies) met variabelen optreden in wiskundige notatie, programmeertalen en natuurlijke taal. Hierbij komen we een aantal belangrijke begrippen regelmatig tegen.

Bereik van kwantoren

Tussen de formules $(\exists x\, Ax \wedge \forall y\, Bxy)$ en $\exists x\, (Ax \wedge \forall y\, Bxy)$ zit een belangrijk verschil. In de eerste formule 'hoort' $\exists x$ alleen bij Ax, in de tweede formule 'hoort' $\exists x$ bij $(Ax \wedge \forall y\, Bxy)$. We spreken in verband hiermee van het *bereik* van een kwantor. Het bereik van $\exists$ in de eerste formule is Ax, het bereik van $\exists$ in de tweede formule is $(Ax \wedge \forall y\, Bxy)$. Het bereik van $\forall y$ is in beide formules Bxy. In het algemeen is het bereik van een voorkomen van een kwantor in een formule φ de subformule waarvóór die kwantor is gevoegd tijdens de constructie van φ.

Het bereik van een kwantor is bijvoorbeeld van belang in een formule waar een variabele onder het bereik van twee kwantoren lijkt te vallen.

Voorbeeld 6.8

De constructieboom voor $\varphi = \forall x\, (Ax \to \exists y\, (Rxy \wedge \forall x\, Sxy))$ ziet er zo uit:

- $\forall x\, (Ax \to \exists y\, (Rxy \wedge \forall x\, Sxy))$
  - $Ax \to \exists y\, (Rxy \wedge \forall x\, Sxy)$
    - Ax
    - $\exists y\, (Rxy \wedge \forall x\, Sxy)$
      - $Rxy \wedge \forall x\, Sxy$
        - Rxy
        - $\forall x\, Sxy$
          - Sxy

Het bereik van de eerste $\forall$ is $(Ax \to \exists y\, (Rxy \wedge \forall x\, Sxy))$, het bereik van de tweede $\forall$ is Sxy.

Vrije en gebonden variabelen

Een voorkomen van een variabele x heet *vrij* als het niet in het bereik van een kwantor $\forall x$ of $\exists x$ ligt. De kwantorvariabelen zelf tellen we hier niet mee. Een kwantor $\forall x$ of $\exists x$ *bindt* de voorkomens van variabelen x in zijn bereik, tenminste, voorzover die nog vrij zijn. Als in een formule φ de variabele x één of meer keer vrij voorkomt en we maken een nieuwe formule door $\forall x$ of $\exists x$ voor φ te zetten, dan raakt x *gebonden*.

Voorbeeld 6.9

- In $\neg(Ax \wedge Rxy)$ komen x en y vrij voor. In $\forall x \neg(Ax \wedge Rxy)$ is x gebonden, y niet. In $(\forall x \neg(Ax \wedge Rxy) \rightarrow \exists y\ (Syx \wedge Bz))$ bindt $\forall x$ de variabele x in $\neg(Ax \wedge Rxy)$ en bindt $\exists y$ de variabele y in $Syx \wedge Bz$. Maar in deze formule is x vrij in $Syx \wedge Bz$, evenals y in $(Ax \wedge Rxy)$. De variabele z is vrij in $(\forall x \neg(Ax \wedge Rxy) \rightarrow \exists y\ (Syx \wedge Bz))$. Merk op dat de plaatsing van de haakjes belangrijk is: in $\forall x\ (\neg(Ax \wedge Rxy) \rightarrow \exists y\ (Syx \wedge Bz))$ wordt elke x gebonden door $\forall x$.

Voorbeeld 6.10

Binden komt ook elders veel voor:

- $\sum_{i=1}^{5} a_{ij}$ (*i* gebonden, *j* vrij)
- $\int_{-1}^{1} \sin(x + y)dx$ (*x* gebonden, *y* vrij)
- FOR $x = 1$ TO 4 DO $y := y \cdot x$; $z := z + 1$ (*x* gebonden, *y* en *z* vrij)

Overigens is de structuur van bindingen in programmeertalen doorgaans veel complexer dan in de predikaatlogica. Hierover meer in hoofdstuk 15 over de semantiek van programmeertalen.

Vrije variabelen

De verzameling vrije variabelen van een formule valt te beschouwen als de verzameling 'parameters' van de door die formule uitgedrukte bewering. De verzameling vrije variabelen wordt met een recursie op termen en formules inductief gedefinieerd:

DEFINITIE 6.4

Vrije variabele
De verzameling *vrije variabelen* van een term t of formule φ (notatie: $VV(t)$ respectievelijk $VV(\varphi)$) wordt als volgt bepaald:

a $VV(x) = \{x\}$, als x een variabele is; $VV(a) = \varnothing$, voor constanten a
b $VV(f^k(t_1, \ldots, t_k)) = VV(t_1) \cup \ldots \cup VV(t_k)$
c $VV(P^k(t_1, \ldots, t_k)) = VV(t_1) \cup \ldots \cup VV(t_k)$
d $VV(\neg\psi) = VV(\psi)$
e $VV(\psi \wedge \chi) = VV(\psi) \cup VV(\chi)$; evenzo voor $\vee$, $\rightarrow$, $\leftrightarrow$
f $VV(\exists x\, \psi) = VV(\psi) \setminus \{x\}$; evenzo voor $\forall x$

Het effect van deze definitie is dat een variabele x gebonden raakt door de *binnenste* kwantor ($\forall x$ of $\exists x$) in het bereik waarvan x voorkomt.

Gesloten en open formules

We vullen de terminologie nog wat aan: een *zin* of *gesloten formule* is een formule zonder vrije variabelen. Een formule waarin vrije variabelen voorkomen, heet een *open* formule. Pa, $\exists x\ Px$, $\forall x\ \forall y\ (Rxy \vee \exists z\ (Rxz \wedge Rzy))$ zijn voorbeelden van zinnen (a is een individuele constante); Rax, $\exists x\ Sxy$ zijn open formules.

Substitutie

Net als in de propositielogica (hoofdstuk 2) is vervanging van delen van formules door andere uitdrukkingen een nuttig begrip in de predikaatlogica. Bijvoorbeeld, bij een universele bewering $\forall x\ \varphi$ is het zinvol te spreken over de 'speciale gevallen' die hieruit ontstaan door voor x namen van individuen, dat wil zeggen termen, te *substitueren*.

DEFINITIE 6.5

Substitutie

a Laat t en t' termen zijn en x een variabele. Dan is $[t/x]t'$ de term die ontstaat door elk voorkomen van x in t' te vervangen door t.

b Laat φ een formule zijn, t een term en x een variabele. Dan is $[t/x]\varphi$ de formule die ontstaat door elk *vrij* voorkomen van x in φ te vervangen door t. ($[t/x]\varphi$ noemt men wel een *instantie* van φ.)

Voorbeeld 6.11

- $[y/x](\forall x\ (Rx \vee Sx) \vee Pxx) = \forall x\ (Rx \vee Sx) \vee Pyy$

(Alleen de *vrije* voorkomens van x worden vervangen!)

- $[f(a, b)/z]\exists x\ (Px \rightarrow Ryz) = \exists x\ (Px \rightarrow Ryf(a, b))$
- Zij nu echter φ de formule $\exists y\ \ y < x$. Lees φ dus als 'er is een getal kleiner dan (een niet nader gespecificeerd getal) x'. Intuïtief zegt $[y/x]\varphi$ dan: 'er is een getal kleiner dan y', maar wat er in feite ontstaat is dit: $[y/x]\exists y\ y < x = \exists y\ y < y$, hetgeen betekent 'er is een getal kleiner dan zichzelf'!

Vrij voor

De 'substitutie van t voor x' is wel altijd gedefinieerd, maar levert soms ongewenste resultaten op, zoals in het laatste voorbeeld. Blijkbaar dienen substituties vermeden te worden die de structuur der kwantorbindingen veranderen. Om dit te voorkomen, definiëren we wanneer een term t 'vrij voor x' is en daardoor substitueerbaar voor deze variabele zonder ongelukken:

DEFINITIE 6.6

Een term t heet *vrij voor x in φ*, als in $[t/x]\varphi$ geen variabele van t in een gesubstitueerd voorkomen van t gebonden voorkomt.

Voorbeeld 6.12

- f^2zy is niet vrij voor x in $\exists y\ Rxy$, maar wel in Rxy of $\exists u\ Rxu$.
- g^3xyz is vrij voor u in $Pu \vee \forall x\, Pg^3xyz$. Want $[g^3xyz/u](Pu \vee \forall x\ Pg^3xyz) = Pg^3xyz \vee \forall x\, Pg^3xyz$. Hierin komt wel een variabele in een voorkomen van g^3xyz gebonden voor (namelijk x), maar niet in een *gesubstitueerd* voorkomen van g^3xyz.

Alfabetische varianten

In feite zijn substitutieproblemen altijd te ondervangen door over te gaan op zogenaamde *alfabetische varianten* van een formule, die ontstaan door gebonden variabelen door nieuwe te vervangen. Intuïtief gesproken tast zo'n vervanging de betekenis van een formule niet aan: gebonden variabelen zijn abstracte 'plaatsvullers' waarvan de precieze typografische aard irrelevant is.

Voorbeeld 6.13

- Een alfabetische variant van $\varphi = \exists y\ y < x$ is $\exists z\ z < x$. In deze laatste formule is y wel vrij voor x: $[y/x]\exists z\ z < x = \exists z\ z < y$.
- Om verwarring te voorkomen kunnen we de formule $\forall x\ (Ax \rightarrow \exists y\ (Rxy \wedge \forall x\ Sxy))$ uit voorbeeld 6.8 beter schrijven als haar alfabetische variant $\forall z\ (Az \rightarrow \exists y\ (Rzy \wedge \forall x Sxy))$.

6.5 AFSLUITENDE OPMERKINGEN

In dit hoofdstuk is een taal geïntroduceerd die aanleiding zal geven tot het kernsysteem van de 'standaardlogica', namelijk de predikaatlogica. In de komende hoofdstukken zullen allerlei aspecten aan bod komen die we ook bij de propositielogica gezien hebben: semantiek, geldigheidstests en bewijstheorie – alle passend verrijkt vergeleken bij wat we in het vorige blok van dit boek hebben leren kennen. Deze moeite loont. Ook voor de informatica fungeert de predikaatlogica namelijk als een standaardreferentiepunt; het is een geschikt formalisme voor het formuleren van specificaties van programma's, maar ook voor correctheidsbeweringen en andere aspecten van programmagedrag. En ten slotte wordt de predikaatlogica binnen de kunstmatige intelligentie beschouwd als een van de fundamentele mechanismen voor 'kennisrepresentatie'. In de volgende blokken van dit boek zal een en ander nader worden geïllustreerd.

Er zijn vele varianten op de hier gepresenteerde predikaatlogische notatie, ontstaan met het oog op specifieke toepassingen. De syntaxis blijft echter essentieel dezelfde. Deze syntaxis is overigens nogal verschillend van die van de natuurlijke taal, ondanks onze eerdere 'vertaalopgaven'. Een van de interessantste hedendaagse computationele problemen in 'natural language processing' is dan ook om precieze algoritmen te vinden die gewone zinnen in formules omzetten, zodat de computer hiermee dan formeel kan 'rekenen', en vervolgens weer andersom. Enkele aspecten hiervan worden behandeld in hoofdstuk 20.

6.6 OPGAVEN

6.1 Geef de constructieboom voor de formule $(\forall x\, (\forall y\, (Ryx \to Ay) \to Ax) \to \forall x\, (Ax \to \forall y\, (Rxy \to Ay)))$.
Wat is het bereik van de kwantoren? Welke kwantor bindt welke variabele? Geef een alfabetische variant zonder herhalingen van gebonden variabelen voor verschillende kwantoren.

6.2 Welk begrip d wordt gemeten door de volgende inductieve definitie?

$d(\varphi)$	$= 0$, voor atomen φ	
$d(\neg\varphi)$	$= d(\varphi)$	
$d(\varphi \wedge \psi)$	$=$ maximum$(d(\varphi), d(\psi))$	(evenzo voor $\vee, \to, \leftrightarrow$)
$d(\forall x\, \varphi)$	$= d(\varphi) + 1$	(evenzo voor $\exists$)

6.3 Is f^2ag^2xy vrij voor x in de volgende formules?
i $\forall z\, (\forall x\, Ax \wedge Px)$
ii $\forall y\, (Qa \to Px) \to Qx$

6.4 Laat met een tegenvoorbeeld zien dat niet altijd geldt:
$[y/x][x/z]\varphi = [y/z]\varphi$

6.5 Beschouw beweringen over de natuurlijke getallen $\mathbb{N}$. De eenplaatsige functieletter S staat voor de *successor*– of opvolgerfunctie: $Sn = n + 1$.
a Welke beweringen worden uitgedrukt door de volgende formules?
i $\forall x\, \neg(Sx = 0)$
ii $\forall x\, (x = 0 \vee \exists y\; x = Sy)$
iii $\forall x\, \exists y\, (x < y \wedge \neg\exists z (x < z \wedge z < y))$
iv $([0/x]\varphi \wedge \forall x\, (\varphi \to [Sx/x]\varphi)) \to \forall x\, \varphi$ (voor willekeurige φ)
b Schrijf de volgende beweringen met behulp van formules:
i Optellen van getallen is associatief.
ii De som van twee getallen is altijd kleiner dan hun product.
iii Elk even getal groter dan 2 is de som van twee priemgetallen ('Goldbach's Vermoeden').
c Definieer met behulp van formules de volgende begrippen:
i x is even
ii x is een deler van y
iii x is een priemgetal.

6.6 Wat zeggen de volgende formules over getallen?

i $\forall x\, \forall y\, (x < y \rightarrow \exists z\, (x < z \wedge z < y))$

ii $\forall x\, \exists y\, (y \cdot y = x)$

Geef voor elke formule een verzameling getallen waarvoor de formule wel opgaat en één waarvoor de formule niet opgaat.

6.7 Beschouw het domein der *mensen* met daartussen *familierelaties*.

Definieer de volgende predikaten:

$Oxy \Leftrightarrow x$ *is een ouder van* y

$Vx \Leftrightarrow x$ *is een vrouw*

a Schrijf met behulp van O en V als formules:

i Niemand is vader van iedereen

ii Iedereen heeft een moeder

iii Wie een dochter heeft, heeft een zoon.

b Definieer met behulp van O en V de volgende predikaten:

i *grootouder*

ii *zuster*

iii *oom*

6.8 a Geef weer met predikaatlogische formules:

i Er zijn minstens 2 objecten

ii Er zijn precies 2 objecten

iii Precies 2 objecten zijn A

iv Precies 2 A's zijn B's

b De kwantor $\exists!$ wordt als volgt gedefinieerd:

$\exists! x\ \varphi := \exists x\, \forall y\, ([y/x]\varphi \leftrightarrow y = x)$

i Wat betekent dit?

ii Laat met een plaatje zien dat $\exists! x\, \exists! y\, Rxy$ en $\exists! y\, \exists! x\, Rxy$ niet altijd equivalent zijn.

* 6.9 a Geef inductieve definities van substitutie in termen en in formules, en van 't is vrij voor x in φ'.

b Diverse logici hebben notaties voorgesteld voor een predikaatlogica waarin geen variabelen voorkomen. Doe een poging zo'n notatie te ontwerpen, door formules op te vatten als staande voor n–plaatsige predikaten. Hoe leest u dan de kwantoren? Wat voor 'hulpoperatoren' heeft u nodig?