Hoofdstuk 8

Syntaxis van de Predicatenlogica

In de voorgaande hoofdstukken hebben we de propositielogica geïntroduceerd als een formalisme waarmee redeneringen kunnen worden beschreven. Hierbij konden redeneringen worden geanalyseerd tot op het niveau van enkelvoudige beweringen. Een groot aantal redeneringen, ook wiskundige, kan niet met behulp van de propositielogica worden geanalyseerd. Immers, de interne struktuur van de enkelvoudige beweringen wordt buiten beschouwing gelaten. De volgende redenering, die we al in paragraaf 1.2 zijn tegengekomen, kan niet met behulp van de propositielogica worden geanalyseerd:

(8.1) Alle mensen zijn sterfelijk.
Socrates is een mens.
$\therefore$ Socrates is sterfelijk.

Van deze intuïtief geldige redenering zou de vertaling in de propositielogica er als volgt uitzien:

(8.2) p
q
$\therefore\ r$

Het is eenvoudig om na te gaan dat in de propositielogica *niet* geldt $p, q \models r$ (voor propositiesymbolen p, q en r). We kunnen dus concluderen dat de propositielogica niet rijk genoeg is om alle gewenste redeneringen te analyseren. Om deze reden introduceren wij in dit hoofdstuk een rijker logisch formalisme: de *predicatenlogica*. In deze logica worden ook enkelvoudige beweringen ontleed. Het zal blijken dat redenering (8.1) logisch geldig is in de predicatenlogica.

De indeling van dit hoofdstuk is als volgt. In §8.1 geven we een informele inleiding tot de predicatenlogica. Vervolgens wordt in §8.2 de syntaxis behandeld van *eerste-ordetalen*, zoals predicaatlogische talen worden genoemd. Paragraaf 8.3 heeft recursie en inductie als onderwerp. In §8.4 wordt de notie

substitutie gedefinieerd en in §8.5, tenslotte, wordt gedemonstreerd hoe redeneringen in natuurlijke taal kunnen worden vertaald naar de predicatenlogica.

8.1 Predicaten, individuen en kwantoren

Bekijken we nog eens redenering (8.1), en proberen we daarbij de logische structuur ervan te begrijpen, dan lijkt het redelijk om deze structuur als volgt weer te geven (zie ook §1.2):

(8.3) Alle x die M zijn, zijn ook S.

a is M.

$\therefore$ a is S.

Daarbij dient men voor M het woord *mens*, voor S het woord *sterfelijk* en voor a de naam *Socrates* te lezen. In feite duiden de woorden *mens* en *sterfelijk* zogenaamde *eigenschappen* aan. Namen, daarentegen, duiden personen, dieren of dingen aan. We zeggen dat namen *individuen* aanduiden. Eigenschappen worden in de logica ook wel *predicaten* genoemd.

In de eerste regel van (8.3) komt de letter x voor. Deze x is een *variabele* en wordt in combinatie met het woord *alle* gebruikt om de eigenschappen M en S op een bepaalde wijze aan elkaar te koppelen. In termen van verzamelingen kan dit worden uitgedrukt door te zeggen dat de verzameling van individuen met de eigenschap *mens* een deelverzameling is van de verzameling individuen met de eigenschap *sterfelijk*. De tweede regel kan men dan lezen als de constatering dat het individu *Socrates* tot de verzameling individuen met de eigenschap *mens* behoort. Een soortgelijke lezing kan men aan de derde regel toekennen. Door deze wiskundige interpretatie te geven aan de redenering wordt het zonneklaar dat deze geldig is. In figuur 8.1 is een en ander in de vorm van een Venndiagram weergegeven. De ovalen daarin stellen verzamelingen voor en punten elementen daarvan. Uit het diagram kan het volgende worden afgelezen:

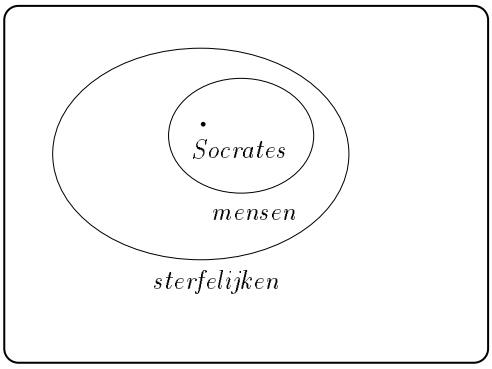
(8.4) $\mathit{mensen} \subseteq \mathit{sterfelijken}$.

$\mathit{Socrates} \in \mathit{mensen}$.

$\therefore$ $\mathit{Socrates} \in \mathit{sterfelijken}$.

Deze wiskundige interpretatie is ook de gangbare, zoals we zullen zien in hoofdstuk 9.

We zijn al een eind op weg bij het bepalen welke ingrediënten de predicatenlogica zou moeten bezitten. Om nog meer greep te krijgen op de gewenste ingrediënten geven we nog een voorbeeld van een geldige redenering:



Figuur 8.1: De 'Socrates-redenering' als Venn-diagram.

(8.5) Alle informatici zijn mensen.

Sommige informatici vinden logica leuk.

$\therefore$ Sommige mensen vinden logica leuk.

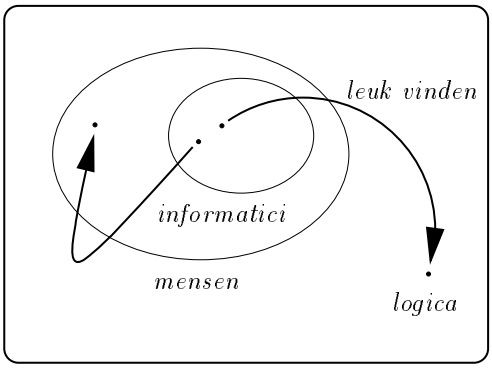
Deze redenering is een stuk lastiger om te doorzien. Met name de tweede en derde regel lijken ingewikkeld. Laten we eerst eens bekijken welke eigenschappen er in het spel zijn. We zien direct in dat *informaticus* en *mens* eigenschappen aanduiden. Hiervoor zullen we respectievelijk de letters I en M reserveren. In de tweede regel komen we *leuk vinden* tegen. Het is geen eigenschap die een individu kan bezitten, maar een *relatie* tussen individuen: de relatie die tussen de individuen x en y bestaat indien individu x individu y leuk vindt. We zullen deze relatie met de letter L aanduiden. Tenslotte is *logica* een naam voor het abstracte ding dat men 'logica' noemt. We zullen hiervoor de letter a gebruiken. De redenering kan dan als volgt worden weergegeven:

(8.6) Alle x die I zijn, zijn ook M.

Er is een x die I is en die in de relatie L tot a staat.

$\therefore$ Er is een x die M is en die in de relatie L tot a staat.

Net als voor de 'Socrates-redenering' kunnen we voor deze redenering een diagram tekenen. Behalve verzamelingen hebben we nu ook wiskundige relaties nodig. In figuur 8.2 hebben we de relatie *leuk vinden* weergegeven door een pijl te tekenen tussen de dingen waartussen deze relatie geldt. De oriëntatie van de pijl geeft daarbij de 'richting' van het leuk vinden aan: x vindt y leuk impliceert namelijk niet automatisch het omgekeerde. De rechter pijl in het diagram geeft dus aan dat een of andere informaticus *logica* leuk vindt, de linker pijl duidt aan dat een andere informaticus een of ander mens leuk vindt.



Figuur 8.2: Het diagram behorende bij de 'logica-redenering'.

In dit voorbeeld hebben we te maken met een relatie tussen twee individuen. Dit behoeft echter niet altijd het geval te zijn. Het is niet moeilijk om voorbeelden van relaties tussen drie of meer individuen te bedenken. Neem bijvoorbeeld de relatie die tussen drie individuen x, y en z bestaat, indien x van y naar z fietst. Al dit soort relaties zullen we in de predicatenlogica gebruiken en *predicaten* noemen. Een predicaat kan dus een eigenschap of een relatie zijn.

Een vraag die we nog moeten beantwoorden luidt: 'Wat is de precieze rol van *alle* en *er is een*?' Deze twee uitdrukkingen noemen we in de logica *kwantoren*. De eerste, *alle*, wordt de *universele kwantor* genoemd. Deze wordt gebruikt om uit te drukken dat voor alle individuen een bepaalde bewering geldig is. De eerste regel van (8.3) dient te worden gelezen als:

(8.7) Voor alle x geldt: als x een M is, dan is x ook S.

De tweede kwantor, *er is een*, noemt men de *existentiële kwantor*. Deze drukt uit dat er een individu bestaat dat een bepaalde eigenschap heeft. Men kan de tweede regel van (8.6) als volgt lezen:

(8.8) Er bestaat een x waarvoor geldt: x is een I en x staat in de relatie L tot a.

Merk op hoe in (8.7) en (8.8) de logische connectieven *als* ... *dan* en *en* te voorschijn zijn gekomen. We willen natuurlijk alle logische connectieven kunnen gebruiken in de predicatenlogica.

In het vervolg zullen we 'x is een M' en 'x staat in de relatie L tot a' noteren als $M(x)$, respectievelijk $L(x, a)$. We leggen er hierbij de nadruk op dat de volgorde van de argumenten in $L(x, a)$ van belang is: $L(a, x)$ betekent iets anders dan $L(x, a)$. Als individu a individu x leuk vindt, hoeft het nog

niet zo te zijn dat x iets in a ziet. Ook voor de kwantoren zullen we een notatie gebruiken: $\forall x$ voor *voor alle x geldt*, en $\exists x$ voor *er bestaat een x waarvoor*. Nu kunnen we de redeneringen (8.1) en (8.5) weergeven in de predicatenlogica. De vertaling van (8.1) ziet er als volgt uit:

(8.9) $\forall x[M(x) \rightarrow S(x)]$
$M(a)$
$\therefore\ S(a)$

Redenering (8.5) kan als volgt in de predicatenlogica worden vertaald:

(8.10) $\forall x[I(x) \rightarrow M(x)]$
$\exists x[I(x) \wedge L(x,a)]$
$\therefore\ \exists x[M(x) \wedge L(x,a)]$

Een aspect van wiskundig redeneren dat we nog niet kunnen beschrijven met behulp van de tot nu toe geïntroduceerde begrippen, is het redeneren over *functies*. Wiskundigen doen uitspraken zoals:

(8.11) De functie $f : \mathbb{N} \rightarrow \mathbb{N}$ is monotoon stijgend.

Deze uitspraak bestaat eigenlijk uit twee delen:

1. f is een functie waarvan zowel het domein als het bereik gelijk is aan $\mathbb{N}$;
2. f is monotoon stijgend.

We kunnen deze uitspraken als volgt weergeven:

(8.12) Voor alle x geldt: als x een natuurlijk getal is, dan is $f(x)$ dat ook;

(8.13) en voor alle x en y geldt: als x en y natuurlijke getallen zijn en als $x \leq y$, dan $f(x) \leq f(y)$.

In deze omschrijving zijn x en y variabelen die voor individuen staan, en is *natuurlijk getal* een eigenschap. Ook de rol van $\leq$ zal duidelijk zijn: het is een relatie. Kenmerkend in (8.12) en (8.13) is het gebruik van de uitdrukking $f(x)$. Uitdrukkingen waarin functies voorkomen die een of meer argumenten hebben, mogen in de predicatenlogica worden gebruikt op dezelfde manier als namen. We kunnen (8.12) en(8.13) als volgt vertalen:

(8.14) $\forall x[N(x) \rightarrow N(f(x))]$,

(8.15) $\wedge\ \forall x \forall y[N(x) \wedge N(y) \wedge K(x,y) \rightarrow K(f(x), f(y))]$.

Hierin staat N voor de eigenschap *natuurlijk getal* en K voor de relatie $\leq$. In de wiskunde wordt deze onhandige notatie waarbij gebruik wordt gemaakt van N en K, niet gebruikt. Men schrijft:

(8.16) $\forall x\,[x \in \mathbb{N} \rightarrow f(x) \in \mathbb{N}]$
$\wedge\ \forall x \forall y [x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge x \leq y \rightarrow f(x) \leq f(y)]$,

of ook wel:

(8.17) $\forall x \in \mathbb{N}\,[f(x) \in \mathbb{N}] \wedge \forall x, y \in \mathbb{N}\,[x \leq y \rightarrow f(x) \leq f(y)]$.

De notatiewijze in (8.17) zullen we niet nader in beschouwing nemen. Over de notatie $x \in \mathbb{N}$ in (8.16) merken we op dat $\in$ hierin een relatie voorstelt en $\mathbb{N}$ een naam van een individu, namelijk de verzameling der natuurlijke getallen. In dat geval is $\mathbb{N}$ dus geen eigenschap.

Het laatste ingrediënt van de predicatenlogica dat we nog moeten beschrijven, is het *gelijkheidssymbool* $=$. In feite is $=$ een speciale relatie waarvan de betekenis op dezelfde wijze is gefixeerd als bijvoorbeeld die van de logische connectieven. Symbolen waarvan de betekenis is gefixeerd, noemt men *logische constanten*. Met behulp van het symbool $=$ kan de uitspraak:

(8.18) Functie $f : \mathbb{N} \rightarrow \mathbb{N}$ heeft een nulpunt.

worden vertaald als:

(8.19) $\forall x [N(x) \rightarrow N(f(x))] \wedge \exists x [N(x) \wedge f(x) = 0]$.

In de bovenstaande vertaling is het symbool 0 een naam van het natuurlijke getal passend bij het begrip nulpunt, nul dus. Merk op dat het consequenter zou zijn om te schrijven $=(f(x), 0)$ in plaats van $f(x) = 0$. Dit is echter hoogst ongebruikelijk en slecht leesbaar.

Als we alle nieuwe mogelijkheden die de predicatenlogica ons biedt, op een rijtje zetten, dan zien we dat we naast de logische connectieven kunnen beschikken over:

- aanduidingen voor predicaten (zowel eigenschappen als relaties), de zogenaamde *predicaatsymbolen*,
- *namen* van individuen, ook wel *individuele constanten* genaamd,
- aanduidingen voor functies, de zogenaamde *functiesymbolen*,
- *variabelen* waarmee individuen worden aangeduid,
- de *universele kwantor* $\forall$ en de *existentiële kwantor* $\exists$, en

- het *gelijkheidssymbool* of *identiteit* $=$.

Logische talen die hierover beschikken, noemt men *eerste-ordetalen*. Deze naamgeving heeft betrekking op het feit dat men kan kwantificeren (met behulp van $\forall$ of $\exists$) over individuen. Individuen worden namelijk beschouwd als eerste-orde-objecten. Voorbeelden van hogere-orde-objecten zijn predicaten en functies. Er bestaan ook hogere-ordetalen. Hierin kan men dus ook kwantificeren over hogere-orde-objecten. Deze talen vallen buiten het kader van dit boek.

8.2 Eerste-ordetalen

Het alfabet van een eerste-ordetaal valt uiteen in een *logisch alfabet* en een *niet-logisch alfabet*. Brengen we in herinnering datgene wat we in de vorige paragraaf hebben gezien, dan bevat de volgende definitie geen verrassingen.

8.2.1 Definitie **Alfabet eerste-ordetaal**
Het alfabet *van een eerste-ordetaal bevat de volgende symbolen:*

1. logische symbolen *(ook* constanten *genaamd):*
   - *(a)* variabelen*:* $x_0, x_1, x_2, \ldots$.
   - *(b)* connectieven*:* $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
   - *(c)* kwantoren*:* $\forall$, $\exists$.
   - *(d)* gelijkheidssymbool *(optioneel):* $=$.
   - *(e) haakjes:* (,) .
   - *(f) komma:* , .
2. niet-logische symbolen*:*
   - *(a)* predicaatsymbolen *(optioneel):* $P_0, P_1, \ldots,$ *waarbij* P_i *een* m_i*-plaatsig predicaatsymbool voorstelt* $(m_i > 0)$.
   - *(b)* functiesymbolen *(optioneel):* $f_0, f_1, \ldots,$ *waarbij* f_j *een* n_j*-plaatsig functiesymbool voorstelt* $(n_j > 0)$.
   - *(c)* namen *(optioneel):* $c_0, c_1, \ldots$.

De bij de predicaatsymbolen en functiesymbolen gebruikte terminologie n-plaatsig *heeft betrekking op het aantal* argumenten *dat het predicaatsymbool of functiesymbool moet hebben.*

	$\mathcal{PL}$	$\mathcal{PL}^{=}$	$\mathcal{PL}^{f}$	$\mathcal{PL}^{f=}$
gelijkheid $=$	neen	ja	neen	ja
predicaatsymbolen $P_0, P_1, \ldots$	ja	ja	ja	ja
functiesymbolen $f_0, f_1, \ldots$	neen	neen	ja	ja
namen $c_0, c_1, \ldots$	ja	ja	ja	ja

Tabel 8.1: De verschillende verschijningsvormen van de predicatenlogica.

Zoals uit de definitie blijkt, behoeft het alfabet van een eerste-ordetaal niet alle soorten symbolen te bevatten. In het alfabet kunnen bijvoorbeeld predicaatsymbolen en namen voorkomen, terwijl functiesymbolen ontbreken. Het is zelfs mogelijk dat zowel predicaatsymbolen, als functiesymbolen en namen totaal ontbreken in het alfabet. Dit is echter een vreemde situatie, zodat we in het vervolg zullen aannemen dat een eerste-ordetaal minstens één predicaat- of functiesymbool bevat. Het alfabet van een eerste-ordetaal bevat echter in elk geval, op het gelijkheidssymbool na, alle logische constanten. Het gelijkheidssymbool $=$ kan voorkomen in het alfabet, maar dat is niet noodzakelijk.

Propositiesymbolen zou men kunnen opvatten als 0-plaatsige predicaatsymbolen, dus als predicaatsymbolen zonder argumenten. Volgens bovenstaande definitie zijn 0-plaatsige predicaatsymbolen echter niet toegestaan. Dit is geen wezenlijke beperking, zodat men de predicatenlogica kan beschouwen als een uitbreiding van de propositielogica. Anders gezegd, de propositielogica is een onderdeel van de predicatenlogica.

8.2.2 Voorbeeld De zuivere predicatenlogica.
De zogenaamde *zuivere predicatenlogica* bevat de volgende symbolen:

- de logische symbolen, behalve het gelijkheidssymbool,
- predicaatsymbolen: $P_0, P_1, \ldots,$
- namen: $c_0, c_1, \ldots.$

De zuivere predicatenlogica bevat dus geen functiesymbolen. In het vervolg zullen we de zuivere predicatenlogica kortweg *predicatenlogica* noemen en aanduiden met $\mathcal{PL}$. Een overzicht van de verschillende vormen van predicatenlogica treft men aan in tabel 8.1. ■

8.2.3 Voorbeeld De taal van de rekenkunde.
Een eerste-ordetaal voor de rekenkunde, waarin optellen en vermenigvuldigen mogelijk is, bevat naast de logische constanten de volgende symbolen:

- het gelijkheidssymbool: $=$.
- de functiesymbolen: $\mathbf{s}$, $\mathbf{+}$, $\cdot$.
- de naam: $\mathbf{0}$.

De bedoelde interpretatie van deze symbolen is (uiteraard) als volgt: $\mathbf{s}$ staat voor de opvolgerfunctie die van een natuurlijk getal de opvolger bepaalt, $\mathbf{+}$ staat voor de operatie optelling, $\cdot$ voor de vermenigvuldiging en $\mathbf{0}$ voor het natuurlijke getal nul. Uit deze omschrijving volgt dat $\mathbf{s}$ een éénplaatsig functiesymbool is en dat $\mathbf{+}$ en $\cdot$ tweeplaatsig zijn. ■

In een eerste-ordetaal onderscheidt men twee syntactische categorieën: die van de *termen* en die van de *formules*. De verzameling van termen wordt genoteerd als *TERM* en de verzameling van alle formules als *FORM*. Beide categorieën worden inductief gedefinieerd in de volgende definities.

8.2.4 Definitie **Term**
De verzameling TERM is de kleinste verzameling die voldoet aan:

1. $x_i \in TERM$ *voor alle* $i \in \mathbb{N}$.
2. $c \in TERM$ *voor alle namen* c.
3. *Als* f *een* n*-plaatsig functiesymbool is en als* $t_1, \ldots, t_n \in TERM$, *dan* $f(t_1, \ldots, t_n) \in TERM$.

De verzameling $\{x_i \mid i \in \mathbb{N}\}$ *van alle variabelen wordt aangeduid met VAR.*

8.2.5 Voorbeeld Termen.

1. Voorbeelden van termen uit de zuivere predicatenlogica zijn: c_0, x_3, c_{13}. Namen en variabelen zijn de enige termen, aangezien de zuivere predicatenlogica geen functiesymbolen heeft.
2. Voorbeelden van termen uit de taal van de rekenkunde zijn: $\mathbf{s}\,(\mathbf{0})$, $\mathbf{s}\,(\mathbf{s}\,(\mathbf{0}))$ (die respectievelijk voor de getallen één en twee staan), $\mathbf{+}\,(x, y)$, $\cdot\,(x, y)$ en $\mathbf{s}\,(\mathbf{+}\,(x, \cdot\,(\mathbf{0}, y)))$. ■

8.2.6 Definitie **Formule**
De verzameling FORM is de kleinste verzameling die voldoet aan:

1. *Als* P *een* n*-plaatsig predicaatsymbool is en* $t_1, \ldots, t_n \in TERM$, *dan* $P(t_1, \ldots, t_n) \in FORM$.
2. *Als* $t_1, t_2 \in TERM$, *dan* $(t_1 = t_2) \in FORM$.

3. *Als* $A, B \in FORM$,
   dan $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B) \in FORM$.

4. *Als* $A \in FORM$ *en* $x \in VAR$, *dan* $\forall x A, \exists x A \in FORM$.

Formules van de vorm $P(t_1, \ldots, t_n)$ *of* $(t_1 = t_2)$ *worden* atomen *of* atomaire formules *genoemd. De verzameling atomen wordt genoteerd als* $ATOM$.

Literalen *zijn formules van de vorm* A *of* $\neg A$ *waarbij* $A \in ATOM$. *De verzameling literalen wordt genoteerd als* LIT.

8.2.7 VOORBEELD Formules.

1. Voorbeelden van formules uit de zuivere predicatenlogica zijn: $P_1(x_3, c_2)$, $(P_1(c_0, x_1) \rightarrow P_4(c_1, c_0, x_2, c_0))$ en $\forall x_0(\neg P_1(x_0, x_3) \wedge \exists x_1 P_4(x_0, x_1, x_3, x_1))$ (aangenomen dat P_1 tweeplaatsig en P_4 vierplaatsig is).

2. Voorbeelden van formules uit de taal van de rekenkunde zijn: $(\mathbf{0} = x_3)$, $\forall x_0 \neg(\mathbf{s}(x_0) = \mathbf{0})$ en $\forall x_0 \forall x_1(\mathbf{+}(x_0, \mathbf{s}(x_1)) = \mathbf{s}(\mathbf{+}(x_0, x_1)))$. ∎

In de propositielogica hebben we de metavariabele $\star$ geïntroduceerd om tweeplaatsige connectieven mee aan te duiden. Deze metavariabele zullen we ook in de predicatenlogica gebruiken. Daarnaast zullen we gebruik maken van de metavariabele Q om kwantoren mee aan te duiden.

8.2.8 DEFINITIE **De metavariabele Q**
Het symbool Q *is een metavariabele waarvan de waarde altijd een kwantor is; met andere woorden* $\mathsf{Q} \in \{\forall, \exists\}$.

8.2.9 DEFINITIE **Subformule en echte subformule**

1. *Een formule* A *is een* subformule *van de formule* F *als één van de vier onderstaande voorwaarden geldt.*

   (a) $A = F$.

   (b) $F = \neg C$ *en* A *is een subformule van* C.

   (c) $F = (C \star D)$ *en* A *is een subformule van* C *of* D.

   (d) $F = \mathsf{Q}xC$ *en* A *is een subformule van* C.

2. *Een formule* A *is een* echte subformule *van de formule* F *als* A *een subformule is van* F *en* $A \neq F$.

Evenals in de propositielogica (zie §2.5), zullen we in de notatie van formules uit $FORM$ sommige haakjes weglaten. Verder noteren we namen, variabelen, functie- en predicaatsymbolen bij voorkeur zonder indices. In de taal van de rekenkunde zullen we de symbolen $\mathbf{+}$ en $\cdot$ tussen hun argumenten plaatsen (de zogenaamde *infixnotatie*). Tenslotte zullen we ook de haakjes [,], {, en } gebruiken.

8.2.10 VOORBEELD Notatieconventies.

1. Namen: $a, b, c, \ldots$.
2. Variabelen: $x, y, z, u, \ldots$.
3. Functiesymbolen: $f, g, h, \ldots$.
4. Predicaatsymbolen: $P, Q, R, \ldots$.
5. $P(a, x) \rightarrow Q(b, a, y, a)$ staat bijvoorbeeld voor:
   $(P_1(c_0, x_1) \rightarrow P_4(c_1, c_0, x_2, c_0))$.
6. $\forall x[\neg P(x, y) \wedge \exists z Q(x, z, y, z)]$ staat bijvoorbeeld voor:
   $\forall x_0(\neg P_1(x_0, x_3) \wedge \exists x_1 P_4(x_0, x_1, x_3, x_1))$.
7. $\forall x \forall y[x + \mathrm{s}(y) = \mathrm{s}(x + y)]$ staat bijvoorbeeld voor:
   $\forall x_0 \forall x_1(+(x_0, \mathrm{s}(x_1)) = \mathrm{s}(+(x_0, x_1)))$.

De kwantoren $\forall$ en $\exists$ bezitten dezelfde prioriteit als het connectief $\neg$. ∎

Tot slot willen we wijzen op de verschillende betekenissen van het $=$-teken. Als symbool van het alfabet van een eerste-ordetaal moet het louter als symbool worden gezien. Als zodanig kan het dan voorkomen in een formule zoals $(x_7 = x_9)$. Een ander gebruik dat we van het symbool $=$ maken, is in metatalige uitspraken als $F = (P(x) \rightarrow Q(x))$. Hiermee wordt uitgedrukt dat de metavariabele F de waarde $(P(x) \rightarrow Q(x))$ heeft, dus dat F gelijk is aan de formule $(P(x) \rightarrow Q(x))$. Men zou voor deze verschillende betekenissen van $=$ verschillende tekens kunnen gebruiken. Dit zullen we echter niet doen. De context maakt steeds duidelijk in welke betekenis het symbool $=$ wordt gebruikt.

8.3 Inductie en recursie

In de propositielogica kan men door middel van structurele inductie over de formules in *PROP* bewijzen dat een eigenschap P geldt voor alle formules in *PROP*. Dit inductiebeginsel kan op vrijwel identieke wijze geformuleerd worden voor de verzamelingen *TERM* en *FORM*.

Evenals in de propositielogica kunnen er op recursieve wijze functies worden gedefinieerd met als domein *TERM* of *FORM* en als bereik een of andere verzameling W.

De volgende twee stellingen hebben deze recursieve definities als onderwerp. Zij garanderen het bestaan van een unieke functie $G : TERM \rightarrow W$ of $G : FORM \rightarrow W$ indien aan bepaalde recurrente betrekkingen met randvoorwaarden is voldaan. Het bewijs van deze stellingen laten we achterwege.

8.3.1 STELLING **Recursieve definities van functies over** *TERM*
Zij gegeven een niet-lege verzameling W en voor ieder n-plaatsig functiesymbool f een functie:

$$H_f : W^n \to W.$$

Dan heeft het stelsel recurrente betrekkingen

$$G(f(t_1, \ldots, t_n)) = H_f(G(t_1), \ldots, G(t_n)) \qquad (t_1, \ldots, t_n \in TERM)$$

met als randvoorwaarde dat voor alle variabelen x en constanten c zowel $G(x)$ als $G(c)$ éénduidig bepaalde elementen van W zijn, precies één oplossing $G : TERM \to W$ die aan alle $t \in TERM$ een unieke functiewaarde $G(t)$ toekent.

8.3.2 STELLING **Recursieve definities van functies over** *FORM*
Zij gegeven een niet-lege verzameling W en een zevental functies:

$$\begin{array}{lcl} H_\neg & : & W \to W, \\ H_\star & : & W \times W \to W, \qquad en \\ H_{\mathsf{Q}} & : & W \times \mathbb{N} \to W. \end{array}$$

Dan heeft het stelsel recurrente betrekkingen

$$\begin{array}{rcl} G(\neg A) & = & H_\neg(G(A)), \\ G(A \star B) & = & H_\star(G(A), G(B)), \\ G(\mathsf{Q}x_i A) & = & H_{\mathsf{Q}}(G(A), i) \qquad (A, B \in FORM \ \ en \ i \in \mathbb{N}), \end{array}$$

met als randvoorwaarde dat $G(P)$ voor alle atomen $P \in ATOM$ een éénduidig bepaald element van W is, precies één oplossing $G : FORM \to W$ die aan alle $F \in FORM$ een unieke functiewaarde $G(F)$ toekent.

In het onderstaande doen wij een beroep op het begrip *machtsverzameling* uit de verzamelingenleer. Voor de lezer die niet vertrouwd is met dit begrip, volgt hier de definitie: de machtsverzameling van een verzameling V, notatie $\mathcal{P}V$, is gedefinieerd als de verzameling van alle deelverzamelingen van V, ofwel $\{W \mid W \subseteq V\}$. De $\mathcal{P}$ staat hier voor de Engelse benaming *powerset* of het Duitse *Potenzmenge*.

Als voorbeeld van een recursief gedefinieerde functie beschouwen we in de onderstaande definitie de functie $VV : TERM \cup FORM \to \mathcal{P}\,VAR$, die van een gegeven term of formule de verzameling van *vrije variabelen* bepaalt. Dit zijn variabelen die niet bij een kwantor horen. In de formule $\forall x A(x, y) \to \exists z B(x, z)$, bijvoorbeeld, hoort de y in $A(x, y)$ bij geen enkele kwantor. Men zegt dat deze variabele y *vrij voorkomt* of een *vrije variabele* is. De x in $A(x, y)$ komt niet vrij voor, want deze hoort bij $\forall x$. Men noemt dit een *gebonden variabele*. Voor de variabele x in $B(x, z)$ geldt dat deze vrij voorkomt: deze x wordt niet gebonden door $\forall x$. Vanwege de prioriteitsregels hoort $\forall x$ alleen bij $A(x, y)$. Men noemt $A(x, y)$ het *bereik* van de kwantor $\forall x$. De variabele z wordt gebonden door $\forall z$ aangezien $B(x, z)$ in het bereik van $\forall z$ voorkomt. Dit alles betekent dat er in

$\forall x A(x,y) \rightarrow \exists z B(x,z)$ twee vrije variabelen voorkomen: y en x. De begrippen *vrij* en *gebonden* hebben dus betrekking op een *voorkomen* (optreden) van een variabele en niet op de variabele zelf. Merk op dat we $\forall x$ en $\forall z$ kwantoren hebben genoemd (in plaats van alleen de $\forall$). Dit is gebruikelijk aangezien een kwantor altijd optreedt in combinatie met een variabele.

8.3.3 DEFINITIE **Vrije variabelen**

1. *De verzameling van* vrije variabelen *van een term $t \in TERM$, notatie $VV(t)$, is recursief gedefinieerd door:*

$$\begin{aligned} VV(x_i) &= \{x_i\}, \\ VV(c) &= \emptyset, \\ VV(f(t_1,\ldots,t_n)) &= VV(t_1) \cup \ldots \cup VV(t_n). \end{aligned}$$

2. *De verzameling van* vrije variabelen *van een formule $F \in FORM$, notatie $VV(F)$, is recursief gedefinieerd door:*

$$\begin{aligned} VV(P(t_1,\ldots,t_n)) &= VV(t_1) \cup \ldots \cup VV(t_n), \\ VV(t_1 = t_2) &= VV(t_1) \cup VV(t_2), \\ VV(\neg A) &= VV(A), \\ VV(A \star B) &= VV(A) \cup VV(B), \\ VV(\mathsf{Q} x_i A) &= VV(A) \setminus \{x_i\}. \end{aligned}$$

Het is duidelijk dat door bovenstaande recurrente betrekkingen met randvoorwaarden een functie $VV : TERM \cup FORM \rightarrow \mathcal{P}\,VAR$ wordt gedefinieerd.

8.3.4 DEFINITIE **Gesloten termen en formules**
Een term $t \in TERM$ wordt een gesloten *term genoemd, indien $VV(t) = \emptyset$. Evenzo wordt een formule $F \in FORM$ een* gesloten *formule genoemd, indien $VV(F) = \emptyset$. Een gesloten formule wordt een* (vol)zin *genoemd. Termen of formules die niet gesloten zijn worden* open *genoemd. De verzameling van gesloten termen zullen we in het vervolg aanduiden met $GTERM$ en die van volzinnen met ZIN.*

8.3.5 VOORBEELD Vrije en gebonden variabelen.

1. De term $t = f(x,a)$ bevat de vrije variabele x, dus $VV(t) = \{x\}$.

2. $VV(\forall x P(x,y)) = \{y\}$. De variabele x is gebonden.

3. $VV(\forall x \exists y P(x,y) \rightarrow P(x,z)) = \{x,z\}$. Let op: de variabele x in $P(x,z)$ is niet gebonden en dus vrij. ■

In het laatste voorbeeld hierboven kwam de variabele x zowel gebonden als vrij voor in één formule. Aangezien dit verwarring kan geven, verdient het aanbeveling om dit zoveel mogelijk te vermijden door in formules de gebonden variabelen anders te kiezen dan de vrije. Dit verandert de betekenis van de formule niet, zoals we in stelling 10.2.7 zullen zien. De intuïtie hierachter is dat gebonden variabelen slechts plaatsen in een formule aan een kwantor binden.

8.4 Substitutie

Vaak is het nodig termen te substitueren voor variabelen in termen of formules. Als s en t termen zijn en x een variabele, dan duidt $s[x:=t]$ het resultaat aan van de vervanging van alle voorkomens van x in de term s door de term t. Hierbij is het toegestaan dat $x \in VV(t)$. Op analoge wijze, als F een formule is, dan duidt $F[x:=t]$ het resultaat aan van de vervanging van alle vrije voorkomens van x door de term t. De resultaten van deze substituties zijn respectievelijk weer een term en een formule.

8.4.1 Definitie **Substitutie in termen**
Zij $t \in TERM$ en x een variabele. De substitutie-*operatie $[x:=t]$ is als volgt recursief gedefinieerd voor termen:*

$$\begin{aligned} y[x:=t] &= \begin{cases} t & \text{als } y = x, \\ y & \text{als } y \neq x, \end{cases} \\ c[x:=t] &= c, \\ f(t_1,\ldots,t_n)[x:=t] &= f(t_1[x:=t],\ldots,t_n[x:=t]). \end{aligned}$$

Stelling 8.3.1 garandeert dat voor iedere variabele x en term t de functie $[x:=t]$: $TERM \rightarrow TERM$ uniek is bepaald.

Substitutie in formules is een delicate aangelegenheid. Dit heeft met name betrekking op substituties in formules van de vorm $\mathsf{Q}yA$. Laten we de verschillende gevallen die zich kunnen voordoen, eens bekijken:

1. Wat zou men bedoelen met $(\forall x P(x,y))[y:=f(z)]$? Het antwoord ligt voor de hand: vervang in de formule $\forall x P(x,y)$ overal y door de term $f(z)$. Dit levert $\forall x P(x,f(z))$.

2. De substitutie $(\forall x P(x,y))[y:=f(x)]$ is problematisch. Directe substitutie als in het vorige geval zou opleveren $\forall x P(x,f(x))$. Het probleem schuilt

in het feit dat de x in $f(x)$ na substitutie gebonden wordt door $\forall x$. Dit is onwenselijk daar in de oorspronkelijke formule $\forall x P(x, y)$ een dergelijke binding tussen de x en de y niet bestond.

Om het probleem op te lossen doen we het volgende: eerst *herbenoemen* we de variabele x in $\forall x P(x, y)$, en daarna voeren we pas de substitutie uit. Als we x herbenoemen als w, dan is het resultaat van de substitutie $\forall w P(w, f(x))$.

Dit herbenoemen is volkomen legaal omdat de naamgeving van gebonden variabelen er niet toe doet. De formules $\forall x P(x, y)$ en $\forall w P(w, y)$ hebben dezelfde betekenis (zie ook hoofdstuk 10, stelling 10.2.7).

3. Het laatste geval is de substitutie $(\forall x P(x, y))[x := f(z)]$. Directe substitutie zou leveren $\forall f(z)\, P(f(z), y)$. Dit is echter geen formule. Ook $\forall x P(f(z), y)$ of $\forall z P(f(z), y)$ zijn geen geschikte kandidaten. Bij nader inzien is het eigenlijk vreemd om een gebonden variabele te willen laten bewerken door een substitutie! Het is immers 'toevallig' dat de gebonden variabele in de voorbeeldformule x heet. Als deze variabele w zou heten, dan zou het resultaat van de substitutie $\forall w P(w, y)$ zijn.

   Aangezien het redelijk is om te eisen dat het resultaat van een substitutie onafhankelijk is van de toevallige keuze van de gebonden variabelen, laten we een substitutie alleen op vrij voorkomende variabelen werken. Het resultaat van de substitutie is dus $\forall x P(x, y)$.

Een formalisering van deze drie gevallen vindt men in dezelfde volgorde terug in het laatste geval van de volgende definitie.

8.4.2 Definitie **Substitutie in formules**
Zij $t \in TERM$ en x een variabele. De substitutie-*operatie $[x := t]$ is als volgt recursief gedefinieerd voor formules:*

Atomaire formules

$$\begin{aligned} P(t_1, \ldots, t_n)[x := t] &= P(t_1[x := t], \ldots, t_n[x := t]), \\ (t_1 = t_2)[x := t] &= (t_1[x := t] = t_2[x := t]). \end{aligned}$$

Niet-atomaire formules

$$\begin{aligned} (\neg A)[x := t] &= \neg(A[x := t]), \\ (A \star B)[x := t] &= (A[x := t] \star B[x := t]), \end{aligned}$$

$$
(\mathsf{Q}yA)[x{:=}t] \quad = \quad \begin{cases}
\mathsf{Q}y(A[x{:=}t]) & \textit{als } y \neq x \textit{ en } [\ y \notin VV(t) \\
 & \quad \textit{of } x \notin VV(A)\ ], \\
\mathsf{Q}z(A[y{:=}z][x{:=}t]) & \textit{als } y \neq x,\ y \in VV(t), \\
 & \quad \textit{en } x \in VV(A), \\
 & \quad \textit{met } z \textit{ de eerste} \\
 & \quad \textit{variabele waarvoor} \\
 & \quad z \notin VV(A) \cup VV(t), \\
\mathsf{Q}yA & \textit{als } y = x.
\end{cases}
$$

De zinsnede 'eerste variabele ...' heeft betrekking op de ordening van de variabelen op hun index: $x_0, x_1, \ldots$.

Door de complexe vorm van de definitie van $(\mathsf{Q}yA)[x{:=}t]$ kan stelling 8.3.2 niet worden aangeroepen om te garanderen dat voor iedere variabele x en term t de functie $[x{:=}t]$: $FORM \to FORM$ uniek is bepaald. Hiertoe is een meer algemene formulering van deze stelling nodig. In de praktijk is echter in te zien dat de definitie 'werkt'.

8.4.3 VOORBEELD Substituties.

1. $(\exists x(x > y))[x{:=}y] = \exists x(x > y)$.

   Niet geldt dat $(\exists x(x > y))[x{:=}y] = \exists y(y > y)$ aangezien x een gebonden variabele is.

2. $(\exists x(x > y))[y{:=}x] = \exists z((x > y)[x{:=}z][y{:=}x]) = \exists z((z > y)[y{:=}x]) = \exists z(z > x)$.

   Niet geldt dat $(\exists x(x > y))[y{:=}x] = \exists x(x > x)$, aangezien de variabele x voorkomt in x. De variabele x wordt eerst herbenoemd tot z. De bovenstaande definitie schrijft voor dat z de eerste variabele is die niet vrij voorkomt in $x > y$ en x. Stelling 10.2.7 drukt echter uit dat de keuze van de variabele semantisch gezien niet uit maakt. De reden dat in de definitie toch een speciale variabele wordt gekozen, is gelegen in het feit dat we de uitkomst van de substitutie-operatie uniek bepaald willen hebben. Anders zou het geen functie zijn. In het vervolg zullen we impliciet aannemen dat de keuze van de nieuwe variabele in overeenstemming is met definitie 8.4.2. ■

Simultane substituties $[x_1{:=}t_1, \ldots, x_n{:=}t_n]$, waarbij n variabelen ($n \geq 2$) tegelijkertijd vervangen worden, kunnen analoog gedefinieerd worden. Aan de hand van voorbeelden zullen we simultane substituties toelichten. Een nauwkeurige definitie zal worden gegeven in paragraaf 16.2.

8.4.4 VOORBEELD Simultane substituties.

1. $f(x,y)[x:=a, y:=b] = f(a,b)$.
2. Een simultane substitutie levert niet altijd hetzelfde resultaat op als de corresponderende herhaalde substitutie, zoals uit het volgende voorbeeld blijkt. Hierin is $[x:=y, y:=x]$ een simultane substitutie en $[x:=y][y:=x]$ de corresponderende herhaalde substitutie.
   (a) $(x=y)[x:=y, y:=x] = (y=x)$.
   (b) $(x=y)[x:=y][y:=x] = (y=y)[y:=x] = (x=x)$. ■

8.5 Vertalen van beweringen

In deze paragraaf zullen we een aantal voorbeelden geven van vertalingen van beweringen in de Nederlandse taal naar een eerste-ordetaal. Bij dit vertaalproces dient de lezer te bedenken dat de natuurlijke taal een veel groter uitdrukkingsvermogen heeft dan een eerste-ordetaal. Immers, een eerste-ordetaal is in de eerste plaats ontworpen voor het beschrijven van wiskundige redeneringen. Dit heeft tot gevolg dat niet-wiskundige beweringen vaak slechts moeizaam vertaald kunnen worden naar predicaatlogische formules.

8.5.1 VOORBEELD Eigenschappen van individuen en relaties tussen individuen.

1. Plato is een wijsgeer: $W(p)$.
2. Marie houdt van Jan: $H(m,j)$.
3. Marie zit tussen Klaas en Jan: $Z(k,m,j)$.
4. Marie houdt meer van Klaas dan van Jan: $M(m,k,j)$. ■

In de volgende voorbeelden wordt voor een aantal uitspraken het vertaalproces stapsgewijs weergegeven. Logische voegwoorden en kwantoren worden daarbij eerst omkaderd en daarna vervangen door de overeenkomstige logische connectieven en kwantoren.

8.5.2 VOORBEELD Vertalingen van beweringen en redeneringen.

1. $\boxed{\text{Alle}}$ schapen zijn wit,
   $\forall x[\ \boxed{\text{als}}\ x$ is een schaap, $\boxed{\text{dan}}\ x$ is wit $]$,
   $\forall x[\ x$ is een schaap $\rightarrow x$ is wit $]$,
   $\forall x[S(x) \rightarrow W(x)]$.

2. $\boxed{\text{Er is een}}$ wit schaap,

   $\exists x[\ x$ is een schaap $\boxed{\text{en}}$ x is wit $]$,

   $\exists x[\ x$ is een schaap $\wedge$ x is wit $]$,

   $\exists x[S(x) \wedge W(x)]$.

   Bedenk dat de voor de hand liggende, maar foute vertaling

   $\exists x[S(x) \rightarrow W(x)]$

   minder uitdrukt dan bedoeld. Deze bewering is namelijk ook waar als er *geen* schapen bestaan. Dat is kennelijk niet de bedoeling. De eerste vertaling is echter alleen waar (en dit is bedoeld!) als er een individu a bestaat dat zowel een schaap, als wit is, dus als $S(a)$ en $W(a)$ beide waar zijn.

3. $\boxed{\text{Alles}}$ trekt alles aan,

   $\forall x[\ x$ trekt $\boxed{\text{alles}}$ aan $]$,

   $\forall x[\forall y(\ x$ trekt y aan $)]$,

   $\forall x[\forall y T(x,y)]$,

   $\forall x \forall y T(x,y)$.

4. $\boxed{\text{Sommige}}$ docenten zijn kinderachtig,

   $\exists x[\ x$ is een docent $\boxed{\text{en}}$ x is kinderachtig $]$,

   $\exists x[\ x$ is een docent $\wedge$ x is kinderachtig $]$,

   $\exists x[D(x) \wedge K(x)]$.

5. $\boxed{\text{Iets}}$ doet me aan iets herinneren,

   $\exists x[\ x$ herinnert me aan $\boxed{\text{iets}}$ $]$,

   $\exists x[\exists y(\ x$ herinnert me aan $y\ )]$,

   $\exists x[\exists y H(x,y)]$,

   $\exists x \exists y H(x,y)$.

6. $\boxed{\text{Sommige}}$ patienten vertrouwen alle doktoren,

   $\exists x[\ x$ is een patient $\boxed{\text{en}}$ x vertrouwt alle doktoren $]$,

   $\exists x[\ x$ is een patient $\wedge$ x vertrouwt $\boxed{\text{alle}}$ doktoren $]$,

   $\exists x[\ x$ is een patient $\wedge$ $\forall y($ $\boxed{\text{als}}$ y is een dokter $\boxed{\text{dan}}$ x vertrouwt $y\ )]$,

   $\exists x[\ x$ is een patient $\wedge$ $\forall y(\ y$ is een dokter $\rightarrow$ x vertrouwt $y\ )]$,

   $\exists x[P(x) \wedge \forall y(D(y) \rightarrow V(x,y))]$.

7. Paardestaarten zijn dierestaarten,

   $\forall x\{$ $\boxed{\text{als}}$ x de staart van een paard is,
   $\boxed{\text{dan}}$ is x de staart van een dier $\}$,

   $\forall x\{$ x is de staart van $\boxed{\text{een}}$ paard $\rightarrow$
   x is de staart van $\boxed{\text{een}}$ dier $\}$,

   $\forall x\{$ $\exists y[$ y is een paard $\boxed{\text{en}}$ x is de staart van y $] \rightarrow$
   $\exists y[$ y is een dier $\boxed{\text{en}}$ x is de staart van y $]\}$,

   $\forall x\{$ $\exists y[$ y is een paard $\wedge$ x is de staart van y $] \rightarrow$
   $\exists y[$ y is een dier $\wedge$ x is de staart van y $]\}$,

   $\forall x\{\exists y[P(y) \wedge S(x,y)] \rightarrow \exists y[D(y) \wedge S(x,y)]\}$. ■

8.6 Opgaven

1. Geef een recursieve definitie van een functie BV die de verzameling van alle gebonden variabelen van een formule bepaalt.

2. Voer de volgende substituties uit:

   (a) $P(x, f(x))[x := f(y)]$.

   (b) $P(x, f(x))[x := f(x)]$.

   (c) $(\forall x P(x,y) \vee \exists y P(x,y))[y := f(z)]$.

   (d) $(\forall x P(x,y) \vee \exists y P(x,y))[y := g(x,y)]$.

3. Vertaal naar de predicatenlogica:

   (a) *Logici kunnen logisch redeneren.*

   (b) *Jan is geen logicus of hij kan logisch redeneren.*

   (c) *You (one) can fool all people sometimes, and you can fool some people all times, but nobody can fool all people all times.*

4. Vertaal de volgende redenering naar de predicatenlogica:

   People are prejudiced against anyone who is liked by someone they dislike.
   But nobody is prejudiced against himself.
   So people don't like anyone who dislikes them.