Hoofdstuk 2

Syntaxis van de Propositielogica

De *propositielogica* houdt zich bezig met de analyse van *proposities* of *beweringen*, en hun logische betrekkingen. Kenmerkend voor deze analyse is het onderscheid tussen *enkelvoudige* en *samengestelde beweringen*, waarbij samengestelde beweringen zijn opgebouwd uit enkelvoudige beweringen en *voegwoorden* of *connectieven* (§2.1). Deze laatste zijn bepalend voor de genoemde logische betrekkingen tussen beweringen.

In de propositielogica worden beweringen voorgesteld als symbolische uitdrukkingen in een *formele taal*. Deze kunstmatige taal noemt men de *taal van de propositielogica*. Bij de beschrijving van zowel natuurlijke als formele talen hanteert men het onderscheid tussen *syntaxis* en *semantiek*. Onder de syntaxis van een taal verstaat men een beschrijving van de grammaticale aspecten ervan. De semantiek, daarentegen, is een beschrijving van de betekenis van de uitdrukkingen in die taal. In §2.2 wordt de syntaxis van de propositielogica behandeld. De semantiek komt aan de beurt in hoofdstuk 3.

De overige paragrafen uit dit hoofdstuk zijn gewijd aan onderwerpen die gerelateerd zijn aan de syntaxis van de propositielogica. In §2.3 komt het begrip *structurele inductie* aan de orde. Dit is een bewijstechniek die verwant is aan volledige inductie over de natuurlijke getallen. Vervolgens behandelen we in §2.4 zogenaamde *recursieve definities*. We besluiten het hoofdstuk met notationele kwesties betreffende de taal van de propositielogica (§2.5).

2.1 Beweringen en connectieven

Zoals gezegd is propositielogica de logica van beweringen en hun logische betrekkingen. Een *bewering* is een abstractie van wat uitgedrukt wordt door een constaterende zin. Voorbeelden van constaterende zinnen zijn:

(2.1) E.W. Dijkstra is een bekend Nederlands informaticus.

(2.2) Twee plus twee is gelijk aan vijf.

(2.3) Socrates is sterfelijk.

(2.4) Socrates is sterfelijk en twee plus twee is gelijk aan vijf.

(2.5) Twee plus twee is niet gelijk aan vijf.

(2.6) Als twee plus twee gelijk is aan vijf, dan is één gelijk aan nul.

(2.7) Mozart was schilder of Frans Brüggen is dirigent.

Al deze zinnen drukken een bepaalde stand van zaken ofwel een constatering uit; vandaar de naamgeving. Let op dat we een onderscheid maken tussen de zin en de bewering die door de zin wordt uitgedrukt. *Constaterende zin* is een grammaticaal concept, terwijl *bewering* een logisch concept is. Het is mogelijk dat verschillende constaterende zinnen dezelfde bewering uitdrukken. Bijvoorbeeld:

(2.8) Socrate est mortel.

en zin (2.3) drukken dezelfde bewering uit. In de praktijk hebben we weinig last van dit onderscheid tussen bewering en constaterende zin, omdat we in dit boek zelden of nooit belang stellen in grammaticale concepten.

Een bewering is *waar* of *onwaar*. Zo zijn bijvoorbeeld de beweringen uitgedrukt door de zinnen (2.1), (2.3), (2.5), (2.6) en (2.7) waar, terwijl de beweringen uitgedrukt door (2.2) en (2.4) onwaar zijn. Het is echter ook gebruikelijk om te spreken over de waarheid of onwaarheid van constaterende zinnen. Een zin is dan waar indien de uitgedrukte bewering overeenkomt met de werkelijke stand van zaken, en anders onwaar. Het zal blijken dat *waarheid* de enige voor de propositielogica relevante eigenschap van beweringen of zinnen is. Andere eigenschappen van beweringen of zinnen, zoals de lengte van een zin, of bijvoorbeeld de diepzinnigheid van een bewering, doen er niet toe.

Hoewel een bewering slechts waar of onwaar kan zijn, is het niet altijd mogelijk uit te maken of een gegeven bewering waar of onwaar is. Beroemde voorbeelden van beweringen waarvan de waarheidswaarde nog steeds niet bekend is, zijn:

(2.9) Het heelal is eindig.

(2.10) *Het vermoeden van Goldbach* (1690 – 1764):

Ieder even natuurlijk getal groter dan vier is de som van twee priemgetallen.

(2.11) *Het laatste theorema van Fermat*[1] (1601 – 1665):

[1]Inmiddels heeft de Engelsman Andrew Wiles dit vermoeden in 1994 bewezen.

> De vergelijking $x^n + y^n = z^n$ met n, x, y en z positieve gehele getallen, heeft voor $n > 2$ geen oplossing.

De grammaticale opbouw van de gegeven zinnen is niet erg ingewikkeld. Zo zijn (2.1), (2.2) en (2.3) enkelvoudige hoofdzinnen die geen voegwoorden bevatten. De zinnen (2.4), (2.5), (2.6) en (2.7), zijn zinnen die uit enkelvoudige hoofdzinnen en voegwoorden zijn samengesteld. De hier gebruikte voegwoorden zijn *en*, *niet*, en *of*, en de 'voegconstructie' *als ... dan*, die we in het vervolg ook als voegwoord zullen beschouwen. Zin (2.4) is bijvoorbeeld samengesteld uit de enkelvoudige hoofdzinnen 'Socrates is sterfelijk' en 'Twee plus twee is gelijk aan vijf', en het voegwoord 'en'. In hoofdstuk 8 zullen we zinnen als 'Socrates is sterfelijk' verder ontleden. Voorlopig zullen we zinnen alleen ontleden in bestanddelen die enkelvoudige constaterende zinnen zijn.

In de volgende paragraaf zullen we een *formele taal* definiëren, waarin we enkelvoudige en samengestelde beweringen kunnen aanduiden. We spreken van een 'formele taal', aangezien het een kunstmatige taal betreft die een mathematische, dus formele, structuur bezit. In die formele taal kunnen we een beperkt aantal voegwoorden aanduiden, te weten *niet*, *en*, *of*, *als ... dan* en *dan en slechts dan als*. Het laatstgenoemde voegwoord treft men vrijwel uitsluitend in wiskundige taal aan.

De genoemde voegwoorden krijgen een exact omschreven betekenis toegewezen die ontleend is aan het gebruik van die voegwoorden in wiskundige taal. De betekenissen die op deze wijze worden verkregen, wijken soms iets af van de betekenissen die de voegwoorden in het Nederlands kunnen bezitten. Omgekeerd kunnen sommige voegwoorden uit het Nederlands niet precies worden weergegeven door die uit de formele taal. De reden hiervoor is dat deze voegwoorden in het normale taalgebruik een te vage betekenis hebben voor nauwkeurig gebruik in de logica.

Aan de hand van een aantal voorbeelden zullen we laten zien dat voegwoorden zoals 'en', 'of', 'niet', 'als ... dan', 'maar', 'omdat' en 'daar' in het normale taalgebruik soms een vage betekenis hebben of zelfs meerdere betekenissen bezitten. Vergelijk om te beginnen het gebruik van het voegwoord 'maar' in de volgende twee zinnen:

(2.12) Cees is een uiterst linkse marxist, maar hij heeft gevoel voor humor.

(2.13) Vijf is een natuurlijk getal, maar vijf is ook een geheel getal.

In zin (2.12) wordt door het gebruik van het voegwoord 'maar' een contrast uitgedrukt, alsof we verbaasd zouden zijn dat Cees ook gevoel voor humor heeft. In zin (2.13) is dit contrast geheel afwezig. Het is ook onduidelijk of we met (2.13) iets anders bedoelen dan met:

(2.14) Vijf is een natuurlijk getal, en vijf is ook een geheel getal.

Uit onderstaande voorbeelden blijkt dat ook voegwoorden als 'en', 'of' en 'als ... dan' in het dagelijkse taalgebruik anders geïnterpreteerd kunnen worden, dan de schrijver of spreker wellicht bedoelde.

(2.15) Als ik gisteren uit het raam van de 13^{e} verdieping was gesprongen, dan was ik gewond geraakt.

(2.16) Als Joost van den Vondel gisteren uit het raam van de 13^{e} verdieping was gesprongen, dan was hij in een vogel veranderd.

(2.17) Als Amsterdam de hoofdstad van Nederland is, dan is zeven een priemgetal.

(2.18) Marie is getrouwd en Marie heeft een baby gekregen.

(2.19) Marie heeft een baby gekregen en Marie is getrouwd.

(2.20) Ruud komt met de auto of Ruud komt te voet.

(2.21) Men kan geld schenken aan het Rode Kruis of aan de Hartstichting.

De beweringen (2.15) en (2.16) hebben dezelfde structuur en de enkelvoudige beweringen waaruit deze beweringen bestaan, zijn alle onjuist. Immers, evenmin als Joost van den Vondel ben ik gisteren uit het raam gesprongen; ook ben ik niet gewond en is Vondel niet in een vogel veranderd. Echter bewering (2.15) zal in het algemeen als waar ervaren worden, terwijl de meeste Nederlandse taalgebruikers bewering (2.16) als onwaar of als onzinnig zullen classificeren. Iets soortgelijks geldt voor bewering (2.17). De beide enkelvoudige beweringen zijn waar, maar omdat elk verband ertussen ontbreekt, wordt de samengestelde 'als ... dan'-bewering als onwaar of zelfs als onzinnig ervaren. Deze voorbeelden tonen aan, dat in het dagelijkse taalgebruik het waar of onwaar zijn van de samengestelde bewering

(2.22) Als p dan q.

niet alleen afhankelijk is van de waarheidswaarden van p en q, maar ook van het verband tussen beide beweringen. Filosofisch is het zeer relevant dit verband tussen p en q te bestuderen en nader te definiëren. Voor het gebruik in de wiskunde en voor veel toepassingen in de informatica is het echter voldoende om de waarheidswaarde van bewering (2.22) slechts te laten afhangen van de waarheidswaarden van p en q en niet van een 'verband' tussen p en q, hoe dan ook gedefinieerd. Men spreekt van de *materiële implicatie*, in tegenstelling tot de *strikte implicatie* waarbij de waarheidswaarde van de bewering afhankelijk is van de waarheidswaarden van p en q, en van de vraag of er een bepaald verband tussen p en q bestaat. Als we de materiële implicatie hanteren, hetgeen we overigens in de rest van dit boek zullen doen, dan moeten we de zinnen (2.15), (2.16) en (2.17) als waar beoordelen. Dit is in overeenstemming met de wiskundige praktijk (vergelijk bijvoorbeeld de zinnen (2.16) en (2.6)). Gebruiken we echter de strikte implicatie dan is alleen zin (2.15) waar.

In de voorbeelden (2.18) en (2.19) suggereert het gebruik van het voegwoord 'en' een tijdsvolgorde. Dit heeft tot gevolg dat deze beweringen, ofschoon ze beide uit dezelfde enkelvoudige beweringen zijn opgebouwd, als enigszins verschillend worden ervaren. In de wiskunde speelt de tijd geen of nauwelijks een rol. In een uitspraak als

(2.23) Het getal e is irrationaal en 5 is een positief geheel getal.

ontbreekt aan het voegwoord 'en' elk tijdsaspect. In de logica zal het voegwoord *en* dan ook geen tijdsaspect bezitten.

De zinnen (2.20) en (2.21) laten zien hoe het voegwoord 'of' in het Nederlands op verschillende manieren kan worden gebruikt. Volgens bewering (2.20) komt Ruud met de auto of te voet, maar niet tegelijkertijd met de auto èn te voet. Bewering (2.21) stelt dat men geld kan schenken aan het Rode Kruis of aan de Hartstichting, maar ook aan beide instellingen. Men zegt dat het voegwoord 'of' in voorbeeld (2.20) *exclusief* wordt gebruikt en in voorbeeld (2.21) *inclusief*. Het voegwoord 'of' wordt in het normale taalgebruik dus afwisselend inclusief en exclusief gebruikt. In de logica zullen we ons echter bedienen van het inclusieve voegwoord *of*.

Resumerend: in de formele taal van de propositielogica zullen we beschikken over de voegwoorden *niet*, *en*, *of*, *als ... dan* en *dan en slechts dan als*. Daarbij is gekozen voor een tijdsonafhankelijk *en*, een inclusief *of* en een materiële implicatie *als ... dan*. Deze keuze is gebaseerd op de wijze waarop de voegwoorden in de wiskunde (en ook in veel andere wetenschappen) worden gebruikt. In plaats van over 'voegwoorden' zullen we voortaan spreken over *connectieven*. *Niet* wordt dan een *éénplaatsig* connectief genoemd, omdat het uit één bewering een nieuwe bewering vormt. De overige connectieven noemt men *tweeplaatsig*, omdat deze uit twee beweringen een nieuwe bewering kunnen vormen.

2.2 De taal van de propositielogica

In deze paragraaf zullen we een formele taal invoeren: de taal van de propositielogica. De 'zinnen' uit deze taal zullen proposities (beweringen) zijn. Een formele taal lijkt in een aantal opzichten op een natuurlijke taal. Zij bezit een *alfabet* waarin de symbolen die in de taal worden gebruikt, zijn vastgelegd, en *grammaticale regels* aan de hand waarvan kan worden vastgesteld of een reeks symbolen een welgevormde uitdrukking van die taal voorstelt. De beschrijving van het alfabet en de grammaticale regels van de taal van de propositielogica vormen samen de *syntaxis* van de propositielogica.

Nu we een taal gaan definiëren, moeten we oppassen dat we niet in de war raken. Immers, het boek dat u nu aan het lezen bent, is óók in een taal geschreven: Nederlands aangevuld met wiskundige taal. De taal van de propositielogica zullen we in dit dialect van het Nederlands beschrijven. De taal van de propositielogica wordt de *objecttaal* genoemd, aangezien zij het 'object' van het beschrijven is. Het met wiskundige taal aangevulde Nederlands waarin die beschrijving plaatsvindt, noemt men de *metataal*.

In het vervolg zullen we de taal van de propositielogica aanduiden als $\mathcal{P}$ ($\mathcal{P}$ropositielogica). In $\mathcal{P}$ worden connectieven aangeduid met symbolen. Enkelvoudige beweringen worden weergegeven door zogenaamde *propositiesymbolen*, waarvoor we $p_0, p_1, \ldots$ zullen gebruiken. Samengestelde beweringen kunnen dan worden geschreven als een combinatie van connectieven, propositiesymbolen en *haakjes*. Deze haakjes hebben dezelfde functie als in wiskundige formules. Dienen ze er in de wiskunde voor om ervoor te zorgen dat de operaties op de juiste termen worden toegepast, in $\mathcal{P}$ zorgen zij ervoor dat het duidelijk is welke proposities door de connectieven met elkaar worden verbonden.

2.2.1 Definitie **Alfabet van $\mathcal{P}$**

Het alfabet van de objecttaal $\mathcal{P}$ is onderverdeeld in drie categorieën die respectievelijk bestaan uit de volgende symbolen:

1. propositiesymbolen: $p_0, p_1, \ldots$.
2. connectieven: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
3. haakjes: (,) .

De connectieven $\neg$, $\wedge$, $\vee$, $\rightarrow$ en $\leftrightarrow$ worden traditioneel met de namen *negatie*, *conjunctie*, *disjunctie*, *implicatie* en *equivalentie* aangeduid (zie ook tabel 2.1). De negatie $\neg$ is een éénplaatsig connectief, terwijl de overige connectieven tweeplaatsig zijn.

Net zomin als de Nederlandse taal bestaat uit alle rijtjes van letters of woorden, bestaat de taal $\mathcal{P}$ uit alle rijtjes van tekens uit het alfabet, zoals gedefinieerd in definitie 2.2.1. Alleen rijtjes die aan bepaalde voorwaarden voldoen —zogenaamd *syntactisch welgevormd* zijn— zijn toegestaan. De syntaxis

connectief	benaming	'betekenis'
$\neg$	negatie	niet
$\wedge$	conjunctie	en
$\vee$	disjunctie	of
$\rightarrow$	implicatie	als ... dan
$\leftrightarrow$	equivalentie	dan en slechts dan als

Tabel 2.1: De connectieven van $\mathcal{P}$.

voorziet in een beschrijving van alle syntactisch welgevormde rijtjes. Om structuur aan te brengen in de syntaxis van de Nederlandse taal, wordt een groot aantal *syntactische categorieën* onderscheiden. Voorbeelden hiervan zijn: de lidwoorden, de zelfstandige naamwoorden, de werkwoorden, de naamwoordelijke zinsdelen en bijwoordelijke bepalingen. In de syntaxis van $\mathcal{P}$ onderscheiden we slechts één syntactische categorie, namelijk die van de *formules.* Een formule is een symboolrij over het alfabet van $\mathcal{P}$ die aan bepaalde voorwaarden voldoet. Aangezien een formule informeel gesproken een enkelvoudige of samengestelde propositie (bewering) aanduidt, zal de verzameling van alle formules worden genoteerd als *PROP* (*PROP*osities). We zullen bovendien de woorden 'propositie' en 'formule' door elkaar gebruiken. Merk op dat we een onderscheid maken tussen de taal $\mathcal{P}$ en de verzameling *PROP* van alle formules van $\mathcal{P}$. Met $\mathcal{P}$ associëren we alle noties die met de propositielogica te maken hebben, waaronder de syntaxis en de semantiek ervan.

2.2.2 DEFINITIE **Formule**
De verzameling PROP is de kleinste verzameling die voldoet aan de volgende eisen:

1. $p_i \in PROP$ *voor alle* $i \in \mathbb{N}$.

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

De elementen van PROP worden formules *genoemd. Verder noemt men formules van de vorm* p_i atomen *en formules van de vorm* p_i *of* $\neg p_i$ literalen.

In bovenstaande definitie worden de letters A en B gebruikt. Deze letters behoren niet tot het alfabet van $\mathcal{P}$. Het zijn variabelen uit de metataal die als *waarde* een formule uit *PROP* hebben. Dit soort variabelen noemt men daarom *metavariabelen.* De uitdrukking '$(A \wedge B)$' duidt de formule aan die bestaat uit een '(', gevolgd door de formule waarvoor A staat, gevolgd door het connectief '$\wedge$', gevolgd door de formule waarvoor B staat, en tenslotte een ')'. Voor metavariabelen die proposities aanduiden, zullen we altijd hoofdletters gebruiken. Deze kunnen eventueel van een index zijn voorzien.

Een geheel andere metavariabele die we in het vervolg vaak zullen gebruiken, is het symbool $\star$.

2.2.3 DEFINITIE **De metavariabele $\star$**
Het symbool $\star$ is een metavariabele waarvan de waarde altijd een tweeplaatsig logisch connectief is; dit wil zeggen $\star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Uit bovenstaande definitie volgt dat de uitdrukking $(A \star B)$ kan staan voor de formules $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ of $(A \leftrightarrow B)$.

2.2.4 VOORBEELD Constructie van een formule.
De symboolrij $F = \neg(p_6 \rightarrow (p_7 \vee p_9))$ is een formule. Dit kan men als volgt inzien: $F_1 = p_6$, $F_2 = p_7$ en $F_3 = p_9$ zijn formules volgens het eerste lid van definitie 2.2.2, en $F_4 = (F_2 \vee F_3) = (p_7 \vee p_9)$, $F_5 = (F_1 \rightarrow F_4) = (p_6 \rightarrow (p_7 \vee p_9))$ en $F_6 = \neg F_5 = \neg(p_6 \rightarrow (p_7 \vee p_9))$ zijn formules volgens het tweede lid van definitie 2.2.2.

De reeks F_1, F_2, F_3, F_4, F_5, $F_6 = F$, ook wel een *constructiereeks* van F genoemd, geeft aan op welke wijze de formule F uit de propositiesymbolen p_6, p_7 en p_9 geconstrueerd kan worden. Merk trouwens op dat de gebruikte F_i $(i = 1, \ldots, 6)$ metavariabelen zijn.

De vraag 'Is F een formule?' kan ook als volgt beantwoord worden. $F = \neg F_5$ is een formule als F_5 een formule is. Op haar beurt is $F_5 = (F_1 \rightarrow F_4)$ een formule als F_1 en F_4 formules zijn. F_1 is het propositiesymbool p_6, dus een formule. Verder is $F_4 = (F_2 \vee F_3)$ een formule als F_2 en F_3 formules zijn. Maar F_2 en F_3 zijn respectievelijk de propositiesymbolen p_7 en p_9 en derhalve formules. Volgens definitie 2.2.2 is F dus een formule, zodat $F \in PROP$.

Uit het bovenstaande blijkt dat men voor iedere symboolrij F op effectieve wijze en in een eindig aantal stappen kan nagaan of $F \in PROP$. ■

2.2.5 DEFINITIE **Subformule en echte subformule**

1. *Een formule A is een* subformule *van de formule F als één van de drie 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.*

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

2.2.6 VOORBEELD Subformules en echte subformules.
De formules F_1, F_2, F_3, F_4, F_5 en $F_6 = F$ uit voorbeeld 2.2.4 zijn alle subformules van F. Ook geldt dat $F_3 = p_9$ een subformule is van F_3, F_4 en F_5.

De vraag 'Is A een subformule van F?' kan op een effectieve wijze en in een eindig aantal stappen worden beantwoord. Immers, dit is het geval dan en slechts dan als A voorkomt in de constructiereeks van F.

De formules F_1, F_2, F_3, F_4 en F_5 zijn echte subformules van F. Deze komen immers alle in de constructiereeks van F voor, en zijn ongelijk aan F. ∎

De definities 2.2.2 en 2.2.5(1) hebben gemeen dat zij op het eerste gezicht circulair lijken. In de eerstgenoemde definitie wordt de verzameling *PROP* gebruikt om *PROP* zelf te definiëren. De tweede definitie gebruikt het begrip subformule in de definitie van dat begrip. Men spreekt van *recursieve definities*. In paragraaf 2.4 wordt deze techniek uitgebreid besproken. Men komt recursieve definities ook vaak in de informatica tegen, namelijk in de vorm van procedures (of functies) die zichzelf aanroepen. Dit worden *recursieve procedures* (of *functies*) genoemd.

We besluiten deze paragraaf met een voorbeeld waarin we laten zien hoe (eenvoudige) Nederlandse zinnen kunnen worden vertaald in de propositielogica.

2.2.7 VOORBEELD Vertalingen van beweringen.
Om beweringen te vertalen naar de propositielogica omkaderen we eerst het hoofdvoegwoord in een zin. Vervolgens vervangen we het omkaderde voegwoord door het ermee corresponderende logische connectief, en omkaderen we de hoofdvoegwoorden in de deelzinnen die ontstaan zijn. Deze stap herhalen we totdat er geen deelzinnen meer zijn waarin een voegwoord kan worden omkaderd. Tenslotte vervangen we de enkelvoudige deelzinnen door geschikte propositiesymbolen.

1. Toos slaapt $\boxed{\text{en}}$ Coby werkt,

   Toos slaapt $\wedge$ Coby werkt,

   $s \wedge w$.

   Hierbij staat s voor 'Toos slaapt', en w voor 'Coby werkt'. In de volgende voorbeelden zullen we dit soort informatie niet expliciet vermelden.

2. $\boxed{\text{Als}}$ het regent, $\boxed{\text{dan}}$ fiets ik niet,

   Het regent $\rightarrow$ Ik fiets $\boxed{\text{niet}}$, Het regent $\rightarrow$ $\neg$ ik fiets,

$r \rightarrow \neg f$.

3. Cees drinkt $\boxed{\text{dan en slechts dan}}$ koffie $\boxed{\text{als}}$ hij er een koekje of een glaasje cognac bij krijgt,

   Cees drinkt koffie $\leftrightarrow$ (hij krijgt er een koekje bij $\boxed{\text{of}}$ hij krijgt er een glaasje cognac bij),

   Cees drinkt koffie $\leftrightarrow$ (hij krijgt er een koekje bij $\vee$ hij krijgt er een glaasje cognac bij),

   $c \leftrightarrow (k \vee c)$. ∎

2.3 Structurele inductie

Vaak is het nodig om te bewijzen dat alle formules uit *PROP* een bepaalde eigenschap bezitten. Hiertoe gebruikt men zogenaamde *structurele inductie*. Dit betekent dat men eerst bewijst dat propositiesymbolen de eigenschap hebben, en vervolgens bewijst dat een samengestelde formule de eigenschap heeft als de samenstellende subformules die eigenschap bezitten. In deze paragraaf wordt deze bewijsmethode nader uitgelegd.

Eigenschappen van natuurlijke getallen kunnen vaak worden bewezen met volledige inductie. Nemen we bijvoorbeeld de eigenschap

$$E(n): \quad \sum_{j=0}^{n} j = \frac{n(n+1)}{2},$$

dan behoeven we om te bewijzen dat E geldt voor alle natuurlijke getallen, slechts aan te tonen dat:

1. E voor het getal 0 geldt, ofwel dat $E(0)$ waar is. Dit noemt men de *inductiebasis*.

2. Als E voor het natuurlijke getal n geldt, E dan ook geldt voor het getal $n+1$. Anders geformuleerd, als $E(n)$ waar is, moet ook $E(n+1)$ waar zijn. Dit noemt men de *inductiestap*.

De aanname '$E(n)$ is waar' in clausule 2 noemt men de *inductieveronderstelling* of *inductiehypothese*. Soms is het handiger om als inductieveronderstelling de variant '$E(k)$ geldt voor $0 \leq k \leq n$' te kiezen. Als men de inductiebasis en de inductiestap bewezen heeft, dan volgt uit het beginsel van volledige inductie dat E geldt voor alle natuurlijke getallen.

Voor de verzameling $PROP$ geldt een analoog beginsel, namelijk het beginsel van *inductie naar de opbouw van de formules in* $PROP$. Ter onderscheiding van volledige inductie, noemt men deze vorm van inductie ook wel *structurele inductie over* $PROP$. De inductie vindt immers plaats over de structuur (de opbouw) van de formules.

2.3.1 DEFINITIE **Structurele inductie over** $PROP$.
Zij P *een eigenschap van formules uit* $PROP$. *Onder een* bewijs door middel van structurele inductie over $PROP$ *dat* P *geldt voor alle* $F \in PROP$, *verstaat men een bewijs van de volgende clausules:*

1. $P(p_i)$ *geldt voor alle propositiesymbolen* $p_i \in PROP$ $(i \in \mathbb{N})$;

2. *Uit de aanname dat* $P(A)$ *geldt, volgt dat* $P(\neg A)$ *geldt* $(A \in PROP)$;

3. *Uit de aanname dat* $P(A)$ *en* $P(B)$ *gelden, volgt dat* $P((A \star B))$ *geldt* $(A, B \in PROP)$.

Clausule 1 noemt men de inductiebasis *De clausules 2 en 3 vormen samen de* inductiestap. *De aanname in clausule 2 dat* $P(A)$ *geldt, en die in clausule 3 dat* $P(A)$ *en* $P(B)$ *gelden, worden de* inductiehypothesen *genoemd.*

In stelling 2.3.5 wordt bewezen dat inductie over $PROP$ een geldige bewijsmethode is. Dat wil zeggen dat de geldigheid van de drie clausules in definitie 2.3.1 inderdaad garandeert dat de betreffende eigenschap geldt voor alle formules in $PROP$.

2.3.2 VOORBEELD Structurele inductie.
Zij $F \in PROP$. Laten we zeggen dat F de eigenschap P bezit, indien het aantal haakjes in F *even* is. We zullen met behulp van structurele inductie laten zien dat alle formules in $PROP$ deze eigenschap bezitten. In het onderstaande bewijs duiden we met $\#(F)$ het aantal haakjes in F aan.

1. Zij $F = p_i$ voor zekere $i \in \mathbb{N}$, ofwel F is een propositiesymbool. Dan geldt zeker $P(p_i)$, aangezien $\#(p_i) = 0$.

2. Zij $F = \neg A$ voor zekere $A \in PROP$, en neem aan dat $P(A)$ geldt, ofwel dat $\#(A)$ even is (inductiehypothese). Hieruit volgt dat $P(\neg A)$, aangezien $\#(\neg A) = \#(A)$.

3. Zij $F = (A \star B)$ voor zekere $A, B \in PROP$, en neem aan dat zowel $P(A)$ als $P(B)$ geldt, ofwel dat $\#(A)$ en $\#(B)$ beide even zijn (inductiehypothese). Maar dan is $\#((A \star B)) = \#(A) + \#(B) + 2$ ook even, en geldt derhalve $P((A \star B))$.

Hiermee is P met structurele inductie bewezen. ■

In het bewijs van de volgende stelling maken we gebruik van het begrip *complexiteit* van een formule.

2.3.3 Definitie **Complexiteit van een formule**
De complexiteit *van een formule $F \in PROP$, notatie $comp(F)$, is gedefinieerd als het aantal connectieven dat in F voorkomt.*

2.3.4 Voorbeeld Complexiteit van formules.

1. $comp(p_5) = 0$;
2. $comp(((p_7 \vee p_3) \leftrightarrow p_1)) = 2$;
3. $comp(\neg(p_1 \vee (p_4 \vee p_2))) = 3$. ■

2.3.5 Stelling *Zij P een eigenschap van formules uit PROP. Als P met structurele inductie over PROP bewezen kan worden, dan geldt P voor alle formules uit PROP.*

Bewijs Neem aan dat de clausules 1, 2 en 3 van definitie 2.3.1 bewezen zijn. Met behulp van volledige inductie naar $comp(F)$ zullen we dan bewijzen dat P geldt voor alle formules uit $PROP$.

1. Zij $comp(F) = 0$. Hieruit volgt dat F een propositiesymbool is. Dit betekent dat $F = p_i$ voor zekere $i \in \mathbb{N}$. Uit clausule 1 volgt nu direct dat $P(F)$ geldt.
2. Zij $comp(F) = n+1$. De inductiehypothese luidt: De eigenschap P geldt voor alle formules C waarvoor $0 \leq comp(C) \leq n$. Voor F doen zich de volgende twee mogelijkheden voor:
   (a) $F = \neg A$. Dan geldt dat $comp(A) = n$. Uit de inductiehypothese volgt $P(A)$. Maar dan geeft toepassing van clausule 2 dat $P(\neg A)$, ofwel $P(F)$.
   (b) $F = (A \star B)$. Dan geldt dat $comp(A)$, $comp(B) \leq n$. Uit de inductiehypothese volgt $P(A)$ en $P(B)$. Toepassing van clausule 3 levert dan $P((A \star B))$, ofwel $P(F)$.

   Hiermee is $P(F)$ aangetoond en de inductiestap voltooid.

Uit het beginsel van volledige inductie volgt nu dat alle formules $F \in PROP$ de eigenschap P bezitten. ■

Tot slot van deze paragraaf geven we nog een voorbeeld van een toepassing van structurele inductie over *PROP*.

2.3.6 VOORBEELD Structurele inductie.
Zeg dat $F \in PROP$ de eigenschap P bezit, indien F hoogstens $2 \cdot comp(F) + 1$ subformules heeft.

Let op het woord 'hoogstens'. Om eigenschap P te bezitten kan een formule F precies $2 \cdot comp(F) + 1$ subformules hebben, maar het mogen er ook minder zijn. Beschouw bijvoorbeeld $\neg p_0$, $(p_1 \wedge p_2)$ en $(p_1 \vee p_1)$. Al deze formules hebben complexiteit 1. De eerste en de derde formule bezitten ieder slechts 2 subformules, terwijl de tweede er 3 bezit.

Het bewijs dat $P(F)$ geldt voor alle $F \in PROP$ verloopt nu aldus.

1. Zij $F = p_i$ voor zekere $i \in \mathbb{N}$, ofwel F is een propositiesymbool. In dit geval geldt zeker $P(F)$ want het aantal subformules van F is gelijk aan $1 = 2 \cdot comp(F) + 1$.

2. Zij $F = \neg A$, waarbij $A \in PROP$. Veronderstel dat $P(A)$. Dit betekent dat A hoogstens $2 \cdot comp(A) + 1$ subformules bezit. Aangezien $comp(\neg A) = comp(A) + 1$ en een subformule van $\neg A$ ofwel een subformule van A, ofwel $\neg A$ zelf moet zijn, heeft F hoogstens $(2 \cdot comp(A) + 1) + 1 < 2(comp(A) + 1) + 1 = 2 \cdot comp(F) + 1$ subformules. Derhalve geldt $P(F)$.

3. Zij $F = (A \star B)$, waarbij $A, B \in PROP$. Veronderstel dat $P(A)$ en $P(B)$. Dit betekent dat A en B respectievelijk hoogstens $2 \cdot comp(A) + 1$ en $2 \cdot comp(B) + 1$ subformules bezitten. Enerzijds geldt dat $comp((A \star B)) = comp(A) + comp(B) + 1$. Anderzijds geldt voor iedere subformule C van $(A \star B)$:

   (a) C is een subformule van A, of

   (b) C is een subformule van B, of

   (c) $C = (A \star B)$.

   De formule F bezit dus hoogstens $(2 \cdot comp(A) + 1) + (2 \cdot comp(B) + 1) + 1 = 2 \cdot comp(F) + 1$ subformules. Derhalve geldt $P(F)$.

We hebben dus met behulp van structurele inductie laten zien dat $P(F)$ geldt voor alle $F \in PROP$. ■

2.4 Recursieve definities

Functies over de natuurlijke getallen kunnen soms gedefinieerd worden door middel van recurrente betrekkingen. In een recurrente betrekking wordt een functie gedefinieerd in termen van zichzelf. Een standaardvoorbeeld is de definitie van de machtsverheffing $f(n) = a^n$ $(a \in \mathbb{N}^+)$. De recurrente betrekking voor f luidt:

$$f(n+1) \quad = \quad a.f(n) \qquad (n \in \mathbb{N})$$

terwijl $f(0) = 1$ een geschikte randvoorwaarde is. We merken op dat in de bovenstaande betrekking het symbool f zowel links als rechts van het symbool $=$ voorkomt: f wordt dus in termen van zichzelf gedefinieerd. Met behulp van volledige inductie is eenvoudig te bewijzen dat de f die aan deze recurrente betrekking met de gegeven randvoorwaarde voldoet, een functie $f : \mathbb{N} \to \mathbb{N}$ is, waarvoor $f(n) = a^n$ voor alle $n \in \mathbb{N}$.

Een ander bekend voorbeeld is de rij van Fibonnacci:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

Vanaf de derde term geldt dat iedere volgende term gelijk is aan de som van de twee direct eraan voorafgaande termen. Noteren we het n^{e} getal van Fibonnacci als $F(n)$, dan kunnen we de rij als een functie op de natuurlijke getallen beschouwen. Deze functie F wordt door de volgende recurrente betrekking met randvoorwaarden gedefinieerd:

$$\begin{aligned} F(0) &= 1, \\ F(1) &= 1, \\ F(n+2) &= F(n) + F(n+1) \qquad (n \in \mathbb{N}). \end{aligned}$$

Het is wederom eenvoudig om met volledige inductie te bewijzen dat dit recurrente stelsel een unieke oplossing F heeft met $F(0) = 1$, $F(1) = 1$, $F(2) = 2$, $F(3) = 3$, $F(4) = 5$, $F(5) = 8, \ldots$.

De volgende stelling geeft aan op welke manier men met behulp van bepaalde stelsels recurrente betrekkingen plus randvoorwaarden functies over *PROP* op een unieke en éénduidige manier kan definiëren.

2.4.1 STELLING **Recursieve definities van functies over** *PROP*
Zij gegeven een niet-lege verzameling W en een vijftal functies:

$$\begin{aligned} h_\neg &: W \to W, \qquad \textit{en} \\ h_\star &: W \times W \to W. \end{aligned}$$

Dan heeft het stelsel recurrente betrekkingen

$$\begin{aligned} f(\neg A) &= h_\neg(f(A)), \\ f((A \star B)) &= h_\star(f(A), f(B)) \qquad (A, B \in PROP), \end{aligned}$$

met als randvoorwaarde dat $f(p_i)$ voor ieder propositiesymbool p_i $(i \in \mathbb{N})$ een éénduidig bepaald element van W is, precies één oplossing $f : PROP \to W$ die aan alle $F \in PROP$ een unieke functiewaarde $f(F)$ toekent.

Het bewijs van deze stelling, dat tamelijk gecompliceerd is, zullen we hier achterwege laten. In concrete gevallen kan men over het algemeen eenvoudig inzien dat een gegeven recurrent stelsel een unieke oplossing f bezit en kan men voor willekeurige $F \in PROP$ gemakkelijk de functiewaarde $f(F)$ bepalen.

2.4.2 VOORBEELD Lengte van een formule.
Zij gegeven het volgende stelsel recurrente betrekkingen met randvoorwaarden:

$$\begin{aligned} f(p_i) &= 1, \\ f(\neg A) &= f(A) + 1, \\ f((A \star B)) &= f(A) + f(B) + 3. \end{aligned}$$

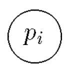
Volgens stelling 2.4.1 heeft dit recurrente stelsel een unieke oplossing f. Iedere functiewaarde van f kan met behulp van de recurrente betrekkingen en de randvoorwaarden worden berekend. Zij bijvoorbeeld $F = (p_1 \wedge (p_2 \to p_3))$, dan kunnen we $f(F)$ als volgt berekenen:

$$\begin{aligned} f((p_1 \wedge (p_2 \to p_3))) &= f(p_1) + f((p_2 \to p_3)) + 3, \\ &= f(p_1) + (f(p_2) + f(p_3) + 3) + 3, \\ &= 1 + 1 + 1 + 3 + 3, \\ &= 9. \end{aligned}$$

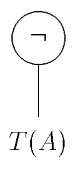
De functie f voegt aan een formule $F \in PROP$ het aantal symbolen dat in F voorkomt, dus de *lengte* ervan, toe. ■

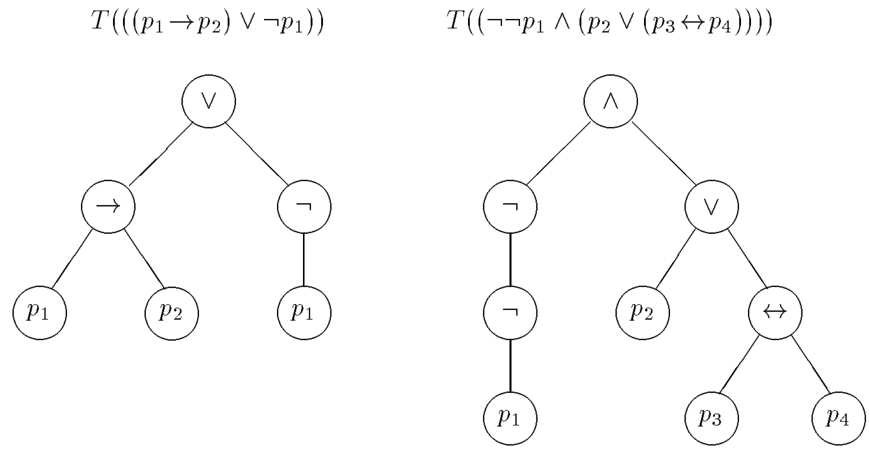
2.4.3 VOORBEELD Ontledingsboom.
De functie T, die recursief wordt gedefinieerd, voegt aan iedere formule $F \in PROP$ haar zogenaamde *ontledingsboom* $T(F)$ toe.

$T(p_i) =$



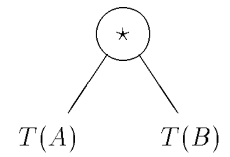
$T(\neg A) =$





Figuur 2.1: Twee ontledingsbomen.

$T((A \star B)) =$



Volgens stelling 2.4.1 is $T(F)$ voor alle $F \in PROP$ op eenduidige wijze gedefinieerd. In figuur 2.1 zijn twee ontledingsbomen weergegeven. ∎

2.5 Notationele kwesties

Voor de menselijke lezer zouden veel haakjes in formules beter weggelaten of door andere soorten haakjes, zoals vierkante haakjes of accoladen, vervangen kunnen worden. Ook de notatie van de propositiesymbolen draagt niet bij tot het leesgemak. Teneinde de leesbaarheid van formules te vergroten, voeren we de volgende notatieconventies in.

1. De propositiesymbolen $p_0, p_1, p_2, \ldots$ worden vervangen door $p, q, r, \ldots$. De symbolen $p, q, r, \ldots$ zijn *metavariabelen* die propositiesymbolen aanduiden.

2. Analoog aan de prioriteitsregels voor de operaties in de rekenkunde ('Meneer Van Dale Wacht Op Antwoord'), zijn er ook precedentieregels voor

prioriteit	connectief
1	$\neg$
2	$\wedge, \vee$
3	$\rightarrow, \leftrightarrow$

Tabel 2.2: De prioriteiten van de connectieven.

de connectieven van $\mathcal{P}$. Deze zijn weergegeven in tabel 2.2. Hierbij geldt dat de prioriteit van een connectief afneemt bij het oplopen van de rangnummers. Merk op dat $\wedge$ en $\vee$, respectievelijk $\rightarrow$ en $\leftrightarrow$ gelijke prioriteiten bezitten. Door toepassing van de gegeven prioriteitsregels kunnen sommige haakjes worden weggelaten. Zo is bijvoorbeeld $\neg p \wedge q \rightarrow r \vee s$ gelijkwaardig aan $((\neg p \wedge q) \rightarrow (r \vee s))$. Immers, de tabel schrijft voor dat de $\wedge$ en de $\vee$ voorafgaan aan de $\rightarrow$. De lezing van formule $p \rightarrow q \leftrightarrow r$ is echter niet eenduidig. Hier moeten haakjes worden geplaatst: $\rightarrow$ en $\leftrightarrow$ hebben een gelijke prioriteit. De formules $p \rightarrow (q \leftrightarrow r)$ en $(p \rightarrow q) \leftrightarrow r$ zijn niet alleen als tekenrij verschillend, ze zijn ook niet onder gelijke omstandigheden waar. De haakjes zijn dus echt noodzakelijk!

3. Niet alleen de haakjes (en) mogen worden gebruikt, ook de rechte haakjes [en], en de accoladen { en } mogen paarsgewijs worden toegepast. Zo mag bijvoorbeeld $((p \rightarrow q) \rightarrow r) \rightarrow s$ worden genoteerd als $[(p \rightarrow q) \rightarrow r] \rightarrow s$.

In het vervolg van dit boek zullen we deze notatieconventies zoveel mogelijk toepassen. We wijzen er met klem op dat bovenstaande notatieconventies zijn ingevoerd voor het gemak van de lezer. Het begrip formule blijft derhalve zoals vastgelegd in definitie 2.2.2. De notaties voor formules die voortkomen door toepassing van deze conventies, dienen te worden beschouwd als *meta-notaties*, dus als metatalige aanduidingen van formules uit de objecttaal.

In de notatie van formules zoals tot nu toe gebruikt, worden de tweeplaatsige connectieven tussen de operanden gezet en de éénplaatsige direct voor de betreffende operand. De haakjes worden gebruikt om aan te geven welke operanden bij een bepaald connectief behoren. Bijvoorbeeld in $p \wedge (q \vee p)$ worden de haakjes om $q \vee p$ gebruikt om aan te geven dat deze subformule een operand is van het connectief $\wedge$.

Met behulp van de zogenaamde *prefix-* of *postfixnotatie* kan het gebruik van haakjes vermeden worden. Bij de prefixnotatie wordt een tweeplaatsig connectief vóór zijn beide operanden en bij de postfixnotatie achter zijn beide operanden gezet. De éénplaatsige operator staat in prefixnotatie vóór en bij postfixnotatie achter zijn operand. Men noemt de prefixnotatie ook wel de *Poolse notatie* en de postfixnotatie de *omgekeerde Poolse notatie*.

2.5.1 VOORBEELD Voorbeelden van prefixnotatie zijn:

1. $\rightarrow p\, q$ in plaats van $p \rightarrow q$;
2. $\vee \wedge p\, q\, r$ in plaats van $(p \wedge q) \vee r$;
3. $\leftrightarrow \wedge \neg\, p \neg\neg\, q \rightarrow p\, q$ in plaats van $(\neg p \wedge \neg\neg q) \leftrightarrow (p \rightarrow q)$. ■

2.5.2 VOORBEELD Voorbeelden van postfixnotatie zijn:

1. $p\, q \rightarrow$ in plaats van $p \rightarrow q$;
2. $p\, q \vee r \wedge$ in plaats van $(p \vee q) \wedge r$;
3. $p \neg\, q \neg\neg \vee p\, q \rightarrow \leftrightarrow$ in plaats van $(\neg p \vee \neg\neg q) \leftrightarrow (p \rightarrow q)$. ■

In de volgende definitie wordt een recursieve definitie gegeven van de prefix- en de postfixnotatie van formules $F \in PROP$.

2.5.3 DEFINITIE **Prefix- en postfixnotatie**
Zij PRE *en* POST *respectievelijk de verzameling van prefixnotaties en de verzameling van postfixnotaties voor de formules uit PROP.*

De functie $\pi : PROP \rightarrow \mathrm{PRE}$ waarvan de functiewaarde $\pi(F)$ gelijk is aan de prefixnotatie van F, kan als volgt recursief worden gedefinieerd:

$$\begin{aligned} \pi(p_i) &= p_i, \\ \pi(\neg A) &= \neg\, \pi(A), \\ \pi((A \star B)) &= \star\, \pi(A)\, \pi(B). \end{aligned}$$

De functie $\Pi : PROP \rightarrow \mathrm{POST}$ waarvan de functiewaarde $\Pi(F)$ gelijk is aan de postfixnotatie van F, kan eveneens recursief worden gedefinieerd:

$$\begin{aligned} \Pi(p_i) &= p_i, \\ \Pi(\neg A) &= \Pi(A)\, \neg, \\ \Pi((A \star B)) &= \Pi(A)\, \Pi(B)\, \star\, . \end{aligned}$$

De functies π en Π zijn gedefinieerd door middel van recursie over *PROP*. Volgens stelling 2.4.1 zijn beide functies dus goed gedefinieerd.

2.5.4 VOORBEELD Prefix- en postfixnotatie.
Met behulp van de definitie van de functies π en Π kan men stapsgewijs de prefix- en postfixnotatie van een gegeven formule F berekenen. Als voorbeeld nemen we de formule $F = (p \rightarrow q) \wedge (r \vee (s \rightarrow t))$.

1. Prefixnotatie.

$$\begin{aligned}\pi(F) &= \wedge\, \pi(p\rightarrow q)\, \pi(r \vee (s\rightarrow t)),\\ &= \wedge\rightarrow p\, q\, \pi(r \vee (s\rightarrow t)),\\ &= \wedge\rightarrow p\, q \vee r\, \pi(s\rightarrow t),\\ &= \wedge\rightarrow p\, q \vee r\rightarrow s\, t.\end{aligned}$$

2. Postfixnotatie.

$$\begin{aligned}\Pi(F) &= \Pi(p\rightarrow q)\, \Pi(r \vee (s\rightarrow t))\wedge,\\ &= p\, q\rightarrow \Pi(r \vee (s\rightarrow t))\wedge,\\ &= p\, q\rightarrow r\, \Pi(s\rightarrow t) \vee \wedge,\\ &= p\, q\rightarrow r\, s\, t\rightarrow \vee \wedge. \qquad \blacksquare\end{aligned}$$

2.6 Opgaven

1. Vertaal de volgende zinnen in de *propositielogica*:
   (a) *Als Jan droomt, dan slaapt hij.*
   (b) *Slapen is een noodzakelijke voorwaarde voor dromen.* (Beschouw hierbij *slapen* en *dromen* als proposities.)
   (c) *Niet drinken is een voldoende voorwaarde om niet dronken te worden.*
   (d) *Als ik gedronken heb en toch autorijd, dan loop ik kans op een forse boete.*
   (e) *Als ik gedronken heb en toch autorijd, dan loop ik kans op een forse boete, behalve als het promillage alcohol in mijn bloed onder een bepaalde waarde ligt.*
   (f) *Als ik autorijd en ik ben dronken of onder invloed van XTC, dan maak ik de weg zeer onveilig.*
2. Geef een recursieve definitie van de functie *comp*.
3. Definieer de *nestingsdiepte* van een formule als het maximum aantal keren dat paren haakjes om een propositiesymbool staan. Zo is bijvoorbeeld de nestingsdiepte van $((p_0 \vee p_1)\rightarrow(p_2 \wedge p_3))$ gelijk aan 2, en die van $((p_0 \vee p_1)\rightarrow(p_2 \wedge (p_3 \vee p_4)))$ gelijk aan 3. Geef een recursieve definitie van de functie *nestingsdiepte*.

4. Bewijs met structurele inductie dat de nestingsdiepte van een formule (zie opgave 3) altijd kleiner dan of gelijk is aan de complexiteit van de formule.

5. Probeer volgens het schema van stelling 2.4.1 een functie te definiëren die voor een formule F de verzameling echte subformules van F oplevert. Waarom lukt dit niet? Pas stelling 2.4.1 zodanig aan dat dit wel mogelijk wordt.