VRIJE
UNIVERSITEIT
BRUSSEL

LOGICA & FORMELE SYSTEMEN

Samenvatting alles vanaf PAGINA 10

29 oktober 2024

Wetenschappen en Bio-ingenieurswetenschappen

Inhoudsopgave

1 Propositielogica 3
1.1 Inleiding 3
1.2 Syntaxis 3
1.2.1 Alfabet 3
1.2.2 Formules 3
1.2.3 Terminologie 3
1.2.4 Formuleschema en instantie 3
1.2.5 Substitutie 4
1.2.6 Constructieboom 4
1.3 Semantiek 4
1.3.1 Waarheidswaarden 4
1.3.2 Waarheidstabellen en waardering 4
1.3.3 Model 5
1.3.4 Tautologie en contradictie 5
1.3.5 Logisch equivalent 6
1.3.6 Functioneel volledig 6
1.3.7 Disjunctieve normaalvorm 7
1.4 Geldig gevolg 7
1.4.1 Geldig gevolg 7
1.4.2 Semantische tableaus 8
1.5 Afleidingen 9
1.5.1 Afleidbaar 9
1.5.2 Syntactisch consistent 10
1.5.3 Stellingen 10
1.5.4 Overzicht afleidingsregels 10
2 Predikaatlogica 11
2.1 Inleiding 11
2.2 Syntaxis 11
2.2.1 Alfabet 11
2.2.2 Plaatsigheid 11
2.2.3 Term 11
2.2.4 Formule 11
2.2.5 Bereik van kwantoren 12
2.2.6 Vrije of gebonden variabelen 12
2.2.7 Gesloten formule 12
2.2.8 Open formule 12
2.2.9 Substitutie 13
2.2.10 Veilige substituties 13
2.2.11 Aflabetische variant 13
2.3 Semantiek 13
2.3.1 Structuur 13
2.3.2 Interpretatiefunctie 14
2.3.3 Model 14
2.3.4 Bedeling 14
2.3.5 Waardering van termen 15
2.3.6 Waardering van formules 15
2.3.7 Gelijkheid van termen 16
2.3.8 Eigenschap waarheidsfunctie 16
2.3.9 Eigenschap substitutie 16
2.3.10 Geldig gevolg 16
2.3.11 Universeel geldig 16

2.3.12 Logisch equivalent 16
2.3.13 Theorie // 17
2.3.14 Axiomaverzameling 17
2.3.15 Modelverzameling 17
2.4 Geldig gevolg 17
2.4.1 Propositielogica vs predikaatlogica 17
2.5 Afleidingen 17
2.6 Metatheorie 18
2.6.1 Substitutie 18
2.6.2 Lemma 19
2.6.3 Prenexvorm 20
2.6.4 Fragmenten van predikaatlogica ???? 20
2.6.5 Adequaatheid van tableaus ???? 20
2.6.6 Volledigheidsstelling ???? 20
3 Lambda calculus 21
3.1.1 Basis begrippen 21
3.1.2 Lambda expressies (syntax) 21
3.1.3 Currying 21
3.1.4 Binding en bereik 21
3.1.5 Vrije variabelen 22
3.1.6 Gesloten lambda-expressie 22
3.1.7 Substitutie: definitie 22
3.2 Rekenen met lambda experssies 22
3.2.1 Beta-gelijkheid 22
3.2.2 Church getallen 23
3.2.3 Lemma (“+”) 23
3.2.4 Lambda-definieerbaar 23

1 Propositielogica

1.1 Inleiding

1.2 Syntaxis

1.2.1 Alfabet

Het alfabet van de propositielogica bestaat uit:

1. een verzameling propositieletters
2. de logische symbolen: $\neg$, $\wedge$, $\vee$, $\rightarrow$ en $\leftrightarrow$
3. de hulpsymbolen:) en (

1.2.2 Formules

De formules van de propositielogica worden als volgt gedefinieerd:

1. elke propositieletter is een formule
2. als φ en ψ formules zijn, dan zijn $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, $(\varphi \leftrightarrow \psi)$ ook formules
3. niets anders is een formule

Uitdrukkingen in de propositielogica noemen we formules. Hiervoor worden griekse letters gebruikt.

Opm: Bovenstaande definitie is een voorbeeld van een inductieve definitie.

Inductieve definitie

Een inductieve definitie heeft 3 kenmerkende onderdelen:

1. een basisstap waarin bepaalde dingen meteen tot objecten van de gewenste soort verklaard worden; *elke propositieletter is een formule*
2. 1 of meer opbouwstappen die verdere constructieprincipes geven om objecten te maken; *als φ en ψ formules zijn, dan zijn $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, $(\varphi \leftrightarrow \psi)$ ook formules*
3. een afsluitende stap die bepaalt dat alles wat niet in eindig veel stappen met behulp van 1 en 2 gevorm kand worden, geen toegestaan object is. *niets anders is een formule*

1.2.3 Terminologie

Propositieletters zijn atomaire formules en worden ook atomen genoemd. Deze zijn niet meer te ontleden in kleinere formules.

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

1.2.4 Formuleschema en instantie

$(\varphi \wedge \psi)$ is een abstracte vorm, geen concrete formule, we noemen dit een formuleschema.

Een instantie van het formuleschema onstaat als we “echte” formules invullen, bv $(p \wedge q)$, $(p \wedge p)$, $((p \rightarrow q) \wedge ((r \rightarrow s) \rightarrow q))$.

1.2.5 Substitutie

Notatie: $[\psi/p]\varphi$
Uitspraak: substitueren van ψ voor p in φ.

Definitie

1. $[\psi/p]\varphi = \psi$ als $\varphi = p$
   $[\psi/p]\varphi = \varphi$ als φ een propositieletter is en $\varphi = p$
2. $[\psi/p]\neg\varphi = \neg[\psi/p]\varphi$
   $[\psi/p](\varphi \wedge \chi) = ([\psi/p]\varphi \wedge [\psi/p]\chi)$
   $[\psi/p](\varphi \vee \chi) = ([\psi/p]\varphi \vee [\psi/p]\chi)$
   $[\psi/p](\varphi \rightarrow \chi) = ([\psi/p]\varphi \rightarrow [\psi/p]\chi)$
   $[\psi/p](\varphi \leftrightarrow \chi) = ([\psi/p]\varphi \leftrightarrow [\psi/p]\chi)$

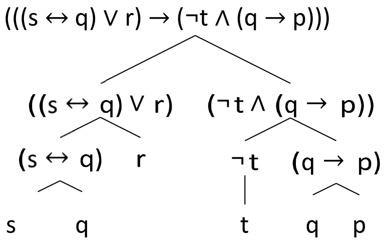
Voorbeeld:

$\psi = (r \wedge s)$
$[\psi/p](p \wedge q) = ((r \wedge s) \wedge q)$

1.2.6 Constructieboom

De opbouw van een formule kan gegeven worden door een constructieboom.

Voorbeeld: $(((s \leftrightarrow q) \vee r) \rightarrow (\neg t \wedge (q \rightarrow p)))$



Opmerkingen:

- Men kan bewijzen dat elke formule exact 1 constructieboom heeft
- Haakjes zijn heel belangrijk
  - Ze geven het bereik aan van connectieven
  - Ze leggen de constructie eenduidig vast
- Zinnen in natuurlijke taal zijn niet eenduidig vast te leggen

1.3 Semantiek

1.3.1 Waarheidswaarden

De betekenis van een formule is zijn waarheidswaarde, waar of onwaar.

Voor atomen kunnen we de waarheidswaarde niet verder analyseren en moeten we waarheidswaarde veronderstellen.

Voor samengestelde formules gebruiken we de waarheidstabellen van de connectieven.

1.3.2 Waarheidstabellen en waardering

φ	$\neg\varphi$
1	0
0	1

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

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

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

Elke rij is een bepaalde situatie. We noemen dit een waardering. Een waardering is een functie van propositieletters naar waarheidswaarden.

Als er n verschillende propositieletters in een formule voorkomen, dan heeft de waarheidstabel van de formule 2^n rijen.
Gevolg: waarheidstabellen kunnen zeer groot worden

Definitie

Een waardering is een functie van alle propositieletters naar waarheidswaarden.

Opmerkingen:

- in de praktijk gebruiken we een beperkt domein: enkel de propositieletters die voorkomen in de formules die we beschouwen
- We gebruiken vaak de functie naam V

1.3.3 Model

Definitie

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

De verzameling van alle modellen van φ noteren we als:

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

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

De verzameling van modellen van Σ noteren we als:

$$MOD(\Sigma)$$

1.3.4 Tautologie en contradictie

Definitie

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

Een formule φ heeft een contradictie als geen enkele waardering een model is van φ, m.a.w. voor elke waardering V geldt: $V(\varphi) = 0$.

Voorbeelden: Alle instanties van

- $(\varphi \vee \neg\varphi)$
- $\neg(\varphi \wedge \neg\varphi)$

zijn tautologiën. Alle

instanties van

- $(\varphi \wedge \neg\varphi)$

zijn contradicties.

1.3.5 Logisch equivalent

Definitie

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

Vaakgebruikte notatie: $\varphi \equiv \psi$

Gevolg Formules φ en ψ zijn logisch equivalent als:

- $(\varphi \leftrightarrow \psi)$ een tautologie is
- Voor elke waardering V geldt $V((\varphi \leftrightarrow \psi)) = 1$
- Voor elke waardering V geldt $V(\varphi) = V(\psi)$

Veelgebruikte logische equivalenties

- φ en $\neg\neg\varphi$
- **de wetten van De Morgan**
  - $\neg(\varphi \wedge \psi)$ en $(\neg\varphi \vee \neg\psi)$
  - $\neg(\varphi \vee \psi)$ en $(\neg\varphi \wedge \neg\psi)$
- **de principes van de distributiviteit**
  - $(\varphi \wedge (\psi \vee \chi))$ en $((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$
  - $(\varphi \vee (\psi \wedge \chi))$ en $((\varphi \vee \psi) \wedge (\varphi \vee \chi))$
- contrapositie: $(\varphi \to \psi)$ en $(\neg\psi \to \neg\varphi)$
- $(\varphi \to \psi)$ en $(\neg\varphi \vee \psi)$
- $(\varphi \leftrightarrow \psi)$ en $((\varphi \to \psi) \wedge (\psi \to \varphi))$

Lossere notatie Volgende formules zijn ook logisch equivalent (associativiteit):

- $(\varphi \wedge (\psi \wedge \chi))$ en $((\varphi \wedge \psi) \wedge \chi)$
- $(\varphi \vee (\psi \vee \chi))$ en $((\varphi \vee \psi) \vee \chi)$

Dit laat toe om haakjes weg te laten, bv. $(((p \wedge q) \wedge r) \wedge s)$ kunnen we ook schrijven als $(p \wedge q \wedge r \wedge s)$. Ook buitentste haakjes worden vaak weggelaten: $p \wedge q \wedge r \wedge s$.

1.3.6 Functioneel volledig

Een verzameling van connectieven C heet functioneel volledig als elke formule φ logisch equivalent is met een formule ψ die enkel connectieven uit C bevat. Stelling: De verzameling $\{\neg, \wedge, \vee\}$ is functioneel volledig.

Gevolg: De verzameling $\{\neg, \vee\}$ is functioneel volledig wegens de tautologie $(\neg(\varphi \wedge \psi) \leftrightarrow (\neg\varphi \vee \neg\psi))$.

1.3.7 Disjunctieve normaalvorm

Definitie

Een formule is in disjunctieve normaalvorm als deze de syntactische vorm heeft van een disjunctie van conjuncties:

$$(\varphi_1 \wedge \ldots \wedge \varphi_{n1}) \vee \cdots \vee (\psi_1 \wedge \ldots \wedge \psi_{nk})$$

waarbij $\varphi_1,\ldots,\varphi_{n1},\ldots,\psi_1,\ldots,\psi_{nk}$ atomen of negaties van atomen zijn.

Voorbeelden:
Disjunctieve normaalvorm:

- $(p \wedge \neg q) \vee (\neg a \wedge b)$
- $a \wedge \neg b \wedge c$
- $a \vee \neg b \vee c$

Geen disjunctieve normaalvorm:

- $\neg(a \wedge b)$
- $a \vee (b \wedge (c \vee d))$

Stelling Voor elke formule φ bestaat er een formule φ^* in disjunctieve normaalvorm zodat φ en φ^* logisch equivalent zijn.

1.4 Geldig gevolg

1.4.1 Geldig gevolg

Definitie

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

Notatie: $\Sigma \vDash \psi$

Alternatieve notatie: de gevolgtrekking Σ/ψ is geldig.

Als $\Sigma = \{\varphi_1,\ldots\varphi_n\}$ schrijven we ook $\varphi_1,\ldots\varphi_n \vDash \psi$

Tautologie Wat als $\Sigma = \emptyset$

- elke waardering is een model van $\emptyset$
- $\vDash \psi$ betekent dat ψ een tautologie is (waar voor elke waardering)

Tegenvoorbeeld Indien ψ niet een geldig gevolg is van Σ noteren we dit als

$$\Sigma \not\vDash \psi$$

Er bestaat dan een model V van Σ dat geen model is van ψ. Het model V heet dan een tegenvoorbeeld van $\Sigma \vDash \psi$.

Voorbeeld: $q, p \to q \not\vDash p$
$V(q) = 1, V(p) = 0$

Geldig gevolg vs waarheid Geldig gevolg en waarheid zijn niet hetzelfde!

$$p \lor q, \neg q \land r \vDash p \land r$$

Maar de waardering $V(p) = V(q) = V(r) = 0$ maakt alle formules onwaar.

Bekende geldige gevolgen

- $\varphi, \neg\varphi \vDash \psi$ voor alle ψ
- contrapositie: $\varphi \to \psi \vDash \neg\psi \to \neg\varphi$
- hypothetisch syllogisme: $\varphi \to \psi, \psi \to \chi \vDash \varphi \to \chi$
- disjunctief syllogisme: $\varphi \lor \psi, \neg\varphi \vDash \psi$

1.4.2 Semantische tableaus

Idee semantische tableaus: we zoeken naar een tegenvoorbeeld. Als we geen tegenvoorbeeld vinden, is het een geldig gevolg.

Sequent

Definitie

Een sequent is een rijtje van de vorm

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

met $\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m$ formules.

Een waardering V heet een tegenvoorbeeld van een sequent $\varphi_1, \ldots, \varphi_n \circ \psi_1, \ldots, \psi_m$ indien $V(\varphi_1) = \ldots = V(\varphi_n) = 1$ en $V(\psi_1) = \ldots = V(\psi_m) = 0$

Een segment $\varphi_1, \ldots, \varphi_n \circ \psi_1, \ldots, \psi_m$ waarbij er een i en j bestaan zodat $\varphi_i = \psi_j$ heeft geen tegenvoorbeelden (belangrijke eigenschap, volgt uit de definitie).

Proces Een semantisch tableau is een schema waarin we op systematische wijze het mogelijke bestaan van tegenvoorbeelden van een gegeven sequent reduceren tot dat van 1 of meer overzichtelijkere sequenten.

Bepalen of $\varphi_1, \ldots \varphi_n / \psi$ een tegenvoorbeeld heeft doen we door een semantisch tableau te maken van $\varphi_1, \ldots \varphi_n \circ \psi$.

Reductieregels Een sequent wordt gereduceerd met behulp van reductieregels.

Notatie: Φ is een rijtje $\varphi_1, \ldots, \varphi_n$, Ψ is een rijtje $\psi_1, \ldots, \psi_m$.

Boom Door het toepassen van reductieregels verkrijgen we een boom. Als er geen reductieregel meer toegepast kan worden spreken we van een semantisch tableau.

Elke tak in een semantisch tableau correspondeert met een route om een tegenvoorbeeld te vinden.

Open/gesloten Een tak heet gesloten indien links en rechts in het sequent dezelfde formule optreedt.

Een tak heet open wanneer de tak niet gesloten is en er geen reductieregels meer toegepast kunnen worden.

Een tableau heet gesloten indien alle takken gesloten zijn. Een tableau heet open indien er minstens 1 tak open is. Dus: geldig gevolg indien een tableau gesloten is.

Semantisch consistent

Definitie

Een formuleverzameling Σ is (semantisch) consistent als Σ een model heeft. Een formuleverzameling die niet consistent is heet inconsistent.

Consistentie via semantisch tableau $\{\varphi_1,\ldots,\varphi_n\}$ is consistent desda er een V bestaat zodat $V(\varphi_1) = \ldots = V(\varphi_n) = 1$.

$\varphi_1,\ldots,\varphi_n \circ$ heeft een tegenvoorbeeld desda er een V bestaat zodat $V(\varphi_1) = \ldots = V(\varphi_n) = 1$.

Dus: $\{\varphi_1,\ldots,\varphi_n\}$ is consistent desda $\varphi_1,\ldots,\varphi_n \circ$ een tegenvoorbeeld heeft (open tableau).

Tautologie via semantisch tableau Een formule φ is een tautologie indien voor elke waardering V geldt dat $V(\varphi) = 1$. Of equivalent indien er geen waardering V bestaat zodat $V(\varphi) = 0$.

$\circ\ \varphi$ heeft een tegenvoorbeeld indien er een waardering V bestaat zodat $V(\varphi) = 0$.

Een formule φ is een tautologie indien er geen tegenvoorbeeld bestaat voor $\circ\ \varphi$ (gesloten tableau).

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

1.5 Afleidingen

In het vorig deel bekeken we geldigheid van een gevolgtrekking op een semantische wijze. Nu gaan we dit op een syntactische manier benaderen.

- bewijzen of afleiden van de conclusie uit de aannames

Natuurlijke deductie Alle regels samen vormen het afleidingssysteem van de natuurlijke deductie.

1.5.1 Afleidbaar

Definitie

Een formule φ heet afleidbaar uit een verzameling aannames Σ als er een afleiding bestaat van φ waarin aan het eind alleen nog aannames uit Σ van kracht zijn.

Notatie: $\Sigma \vdash \varphi$

Indien $\Sigma = \emptyset$, dan heet φ een stelling. Notatie: $\vdash \varphi$ Indien φ niet afleidbaar is uit Σ noteren we $\Sigma \nvdash \varphi$

1.5.2 Syntactisch consistent

Definitie

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

Een verzameling formules die niet consistent is heet inconsistent.

Voorbeelden: Syntactisch consistent: $\{\neg p, p \rightarrow q, q\}$

Syntactisch inconsistent: $\{p, p \rightarrow q, \neg q\}$

1.5.3 Stellingen

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

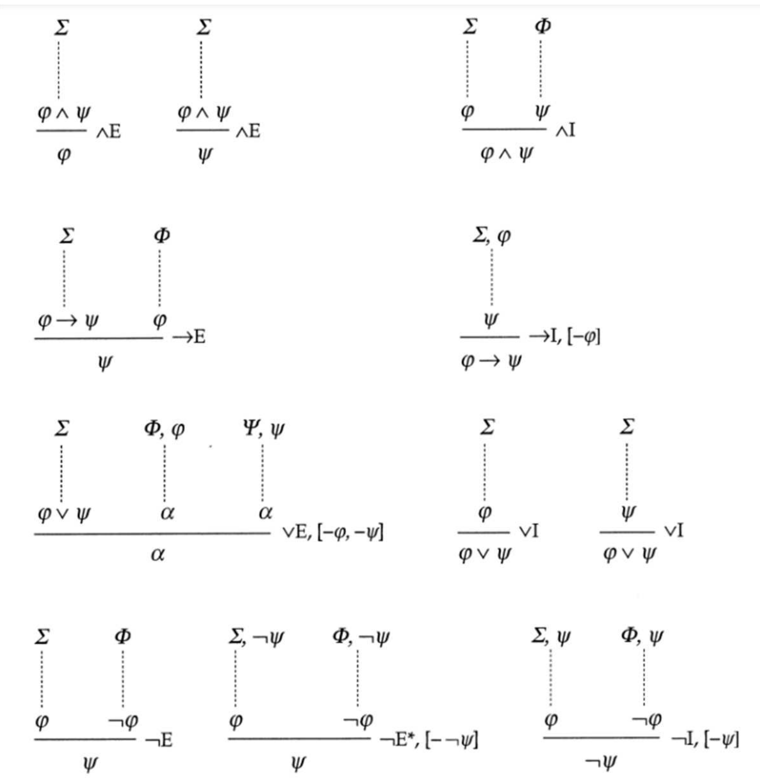
$\Gamma \nvdash \varphi$ desda $\Gamma \cup \{\neg\varphi\}$ is consistent.

Volledigheidsstelling Als Σ een formuleverzameling is en φ een formule, dan geldt: $\Sigma \vdash \varphi$ desda

$\Sigma \vDash \varphi$

- correctheid: als $\Sigma \vdash \varphi$ dan $\Sigma \vDash \varphi$
- volledigheid: als $\Sigma \vDash \varphi$ dan $\Sigma \vdash \varphi$

1.5.4 Overziccht afleidingsregels



2 Predikaatlogica

2.1 Inleiding

2.2 Syntaxis

2.2.1 Alfabet

Het alfabet van de predikaatilogica bestaat uit:

1. een verzameling C van individuele constanten (kleine letters a,b,c…)
2. een verzameling P van predikaatletters (hoofdletters P, Q, R)
3. een verzameling F van functieletters (kleine letters f,g,h…)
4. de logische symbolen: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$ en $\exists$
5. inviduele variabelen (kleine letters u,v,w…)
6. de hulpsymbolen:) en (

2.2.2 Plaatsigheid

Plaatsigheid: aantal argumenten (functieletters, predikaatletters), wordt soms aangeduid via een index

- *F*^3 abc vs *F*(a,b,c) vs *F*^3(a,b,c)

Propositieletters kan je zien als 0-plaatsige predikaatletters.

2.2.3 Term

Definitie

De termen van de predikaatlogica worden als volgt geconstrueerd:

1. Individuele variabelen en consten zijn termen
2. Als f een k-plaatsige functieltter is en $t1, \ldots, tk$ zijn termen dan is $f(t1, \ldots, tk\,)$ook een term
3. Niets anders is een term

Voorbeelden:

+(3,5)

- \+ is een functie
- 3 en 5 zijn termen want constanten

+(3,x)

- \+ is een functie
- 3 is een term want een constante
- X is een term want een variabele

Bij logica gebruiken we prefix notatie

2.2.4 Formule

Definitie

De formules van de predikaatlogica worden als volgt gedefinieerd:

1. als P een k-plaatsige predikaatletter is en $t1, \ldots, tk$ zijn termen, dan is $P(t1, \ldots, tk)$ een formule
2. als φ en ψ formule zijn, dan zijn $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ en $(\varphi \leftrightarrow \psi)$ ook formules
3. als φ een formule is en x een variabele, dan zijn $\forall x\, \varphi$ en $\exists x \varphi$ ook formules
4. niets anders is een formule

Voorbeelden:

Een formule van de vorm is P(t,…tk) met t,…,tk termen heet een atoom of een atomaire bewering

- P^2(x, a)
- P^2(f^2(x, y), g^1(a))

En niet:

- $\forall x(A\text{^}1\ (x\ ) \rightarrow \forall y\ R\text{^}2(x, y) \rightarrow A\text{^}1(y)))$

Zijn geen formules:

- $PP\ \neg xx$
- $A\ x\ \forall x$
- $\forall x \exists R(x, y)$

We proberen notatie zo eenvoudig te houden

- zo weinig mogelijk indices
- zo weinig mogelijk haakjes

2.2.5 Bereik van kwantoren

Het bereik van een kwantor is een formule φ is de subformule waarvoor die kwantor is gevoegd tijdens de constructie van φ

$$(\exists x\ Ax\ \wedge\ \forall y\ Bxy)$$

$$\exists x\ (Ax\ \wedge\ \forall y\ Bxy)$$

2.2.6 Vrije of gebonden variabelen

Definitie

Een voorkomen van een variabele x heet vrij als deze niet in het bereik van een kwantor $\forall x$ of $\exists x$ ligt. Een kwantor $\forall x$ of $\exists x$ bindt de voorkomens van variabelen x in zijn bereik, voor zover deze nog vrij zijn. Deze variabelen heten dan gebonden.

$$\forall x(Ax \rightarrow \exists y(Rxy\ \wedge \forall x\ Sxy))$$

2.2.7 Gesloten formule

Definitie

Een zin of een gesloten formule is een formule zonder vrije variabelen

2.2.8 Open formule

Definitie

Een formule met vrije variabelen heet een open formule

2.2.9 Substitutie

Definitie

1) Laat t en t' termen zijn en x een variabele. Dan is $[t\,/\,x]t'$ de term die onstaat door elk voorkomen van x in t' te vervangen door t.
2) Laat φ een formule zijn, t een term en x een variablele. Dan $[t\,/\,x]\varphi$ de formule die ontstaat door elk vrij voorkomen van x in φ te vervangen door t. $[t\,/\,x]\varphi$ noemt men ook instantie van φ.

Voorbeelden:

2.2.10 Veilige substituties

Definitie

Een term t heet vrij voor x in φ als in $[t\,/\,x]\varphi$ geen variabele van t gebonden wordt.

2.2.11 Aflabetische variant

Een alfabetische variant van een formule onstaat door gebonden variabelen door nieuwe variabelen te vervangen.

Voorbeelden:

- $\exists y\ R\mathrm{xy}$ en $\exists u\ R\mathrm{xu}$
- $\forall x(A\mathrm{x} \rightarrow \exists y(R\mathrm{xy} \wedge \forall x\ S\mathrm{xy}))$ en $\forall z(Az \rightarrow \exists y(Rzy \wedge \forall x\ Sxy))$

Alfabetische varianten worden gebruikt om

- substitutie veilig te kunnen uitvoeren,
- verwarring te verkomen.

2.3 Semantiek

2.3.1 Structuur

Definitie

Een structuur $\boldsymbol{D}$ is een drietal D, $\boldsymbol{R}$, $\boldsymbol{O}$ bestaande uit een niet- lege verzameling DD (het domein), een verzameling $\boldsymbol{R}$ van relaties op DD en een verzameling $\boldsymbol{O}$ van operaties op DD.

Relationele structuur: een structuur met $\mathbf{R} \neq \emptyset$.

- bv D = 1,2,3 met 1 relatie kleiner dan

Operationele structuur: een structuur met $\boldsymbol{O} \neq \emptyset$.

- bv groepen, ringen (algebra)

Voorbeeld:

$$\langle \mathbb{N}, \{<\}, \{0, +, \cdot\} \rangle$$

(met **0** de nul van de natuurlijke getallen -> speciaal object, nul plaatsige operatie)

$\langle D, \boldsymbol{R}, \boldsymbol{O} \rangle$: domein, relaties en operaties

2.3.2 Interpretatiefunctie

Definitie

Laat $\mathbf{D} = D, \boldsymbol{R}, \boldsymbol{O}$ een structuur zijn. Een interpretatiefunctie I kent aan

- elke individuele constante c een speciaal object $I\ c \in \boldsymbol{O}$ toe (nulplaatsige operatie)
- elke predikaatletter P een relatie $I(P) \in \boldsymbol{R}$ toe (van zelfde plaatsigheid)
- elke functieletter f een operatie $I\ f \in \boldsymbol{O}$ (van zelfde plaatsigheid)

Voorbeeld:

$$\langle \mathbb{N}, \{<\}, \{0, +, \cdot\} \rangle$$

Stel P is een 2-plaatsige predikaatletter, f is een 2-plaatsige functieletter, g is een 2-plaatsige functieletter en a is een constante.

voorbeeld interpretatiefunctie

- $I(P) = <$
- $I(f) = +$
- $I(g) = \cdot$
- $I(a) = 0$

2.3.3 Model

Definitie

Een paar $(\boldsymbol{D}, I)$ met $\boldsymbol{D}$ en structuur en I een interpretatiefunctie heet een model.

Een (on)eindig model is een model met een (on)eindig domein.

Voorbeeld:

$M = (\boldsymbol{D}, I)$ waar
$\boldsymbol{D} = \langle \mathbb{N}, \{<\}, \{0, +, \cdot\} \rangle$ en I gedefinieerd als $I(P) = <,\ I(f) = +,$
$I(g) = \cdot,\ I(a) = 0$

$$Px_1x_2 \wedge Pf(a, x_9)g(x_5, x_9)$$

wordt dan geïnterpreteerd als

$$\text{'}x_1 < x_2 \text{ en } x_9 < (x_5 \cdot x_9)\text{'}$$

2.3.4 Bedeling

Definitie

Een bedeling b is een functie die aan elke variabele x een object $b\ x \in D$ toekent.

Voorbeeld:

$M = (\boldsymbol{D}, I)$ waar
$\boldsymbol{D} = \langle \mathbb{N}, \{<\}, \{0, +, \cdot\} \rangle$ en I gedefinieerd als $I(P) = <, I(f) = +,$
$I(g) = \cdot, I(a) = 0$

Voorbeeld bedeling $b(x_i) = i\ (i = 1,2,3, \dots)$

$$Px_1x_2 \wedge Pf(a, x_9)g(x_5, x_9)$$

wordt dan geïnterpreteerd als

$$\text{'}1 < 2 \text{ en } 9 < 45\text{'}$$

Notatie:

- $\langle \mathbb{N}, \{<\}, \{0, +, \cdot\} \rangle$ wordt ook als $\langle \mathbb{N}, <, 0, +, \cdot \rangle$ genoteerd, zolang duidelijk is wat de relaties en wat de functies zijn
- $b[x \mapsto d]$ is de bedeling b die d aan de variabele x toekent, ongeacht wat $b(x)$ ervoor was. Stel $b(x_i) = i\ (i = 1,2,3, \dots)$, dan
  - $b[x_1 \mapsto 10](x_1) = 10$
  - $b[x_1 \mapsto 10](x_2) = 2$

2.3.5 Waardering van termen

Definitie

Laat M = ($\boldsymbol{D}$, I) een model zijn en b een bedeling. De semantische waardering V M,b van termen is alsvolgt gedefinieerd:

- V M,b a = I(a) voor constanten a
- V M,b x = b(x) voor variabelen x
- V M,b(f(t1, … , tk)) = I(f) (V M,b(t1), … , V M,b(tk))

Voorbeeld:

$\boldsymbol{M} = (\boldsymbol{D}, I)$ waar
$\boldsymbol{D} = \langle \mathbb{N}, \{<\}, \{0, +, \cdot\}\rangle$ en I gedefinieerd als $I(P) = <, I(f) = +$,
$I(g) = \cdot, I(a) = 0$
bedeling $b(x_i) = i$ $(i = 1,2,3, \dots)$

$V_{M,b}(f(a, x_1)) = I(f)\left(V_{M,b}(a), V_{M,b}(x_1)\right) = I(f)(I(a), b(x_1)) =$
$+(\mathrm{I}(\mathrm{a}), \mathrm{b}(x_1)) = +(0,1) = 1$

2.3.6 Waardering van formules

Definitie

Laat M = ($\boldsymbol{D}$, I) een model zijn en b een bedeling. De waarheidswaarden van formules is alsvolgt gedefinieerd:

- V M,b(P t1, … , tk) = 1 ⇔ I(P)(V M,b t1 , … V M,b tk) geldt
- V M,b ¬φ= 1 ⇔ V M,b φ= 0 en gelijkaardig voor de andere connectieven• V M,b(f(t1, … , tk)) = I(f) (V M,b(t1), … , V M,b(tk))
- VV MM,bb ∃xx $\varphi\varphi$= 1 ⇔ er is een dd ∈ DD zodat VV MM,bb xx↦dd $\varphi\varphi$= 1
- VV MM,bb ∀xx $\varphi\varphi$= 1 ⇔ voor alle dd ∈ DD geldt VV MM,bb xx↦dd $\varphi\varphi$= 1

Notatie:

Het is gebruikelijk om $\boldsymbol{M}, b \vDash \varphi$ te schrijven in plaats van $V_{M,b}(\varphi) = 1$.

Het is gebruikelijk om $\boldsymbol{M}, b \nvDash \varphi$ te schrijven in plaats van $V_{M,b}(\varphi) = 0$.

Als er geen verwarring mogelijk is schrijven we $V(\varphi)$ in plaats van $V_{M,b}(\varphi)$.

Opmerking:

Propositielogica: een model is een waardering die de formule waarmaakt.

Predikaatlogica: een model is een paar $(\boldsymbol{D}, I)$

Toch is het een passende generalisatie. We blijven zeggen dat een paar $\mathrm{M} = (\boldsymbol{D}, I)$ een model van een formule φ is indien voor elke bedeling b geldt $V_{M,b}(\varphi) = 1$.

2.3.7 Gelijkheid van termen

De gelijkheidsrelatie is niet standaard gedefinieerd. Indien nodig moeten we dit expliciet definieren. Voorbeeld definitie:
$V_{M,b}(t_1 = t_2) = 1$ desda $V_{M,b}(t_1) = V_{M,b}(t_2)$

2.3.8 Eigenschap waarheidsfunctie

De waarheidswaarde is enkel afhankelijk van de bedeling van de vrije variabelen. //

2.3.9 Eigenschap substitutie

Voor alle termen t en t' en een variabele x geldt

$$V_{M,b}([t/x]t') = V_{M,b[x \mapsto V_{M,b}(t)]}(t')$$

2.3.10 Geldig gevolg

Definitie

Laat Σ een verzameling formules zijn en ψ een formule. Dan zeggen we dat ψ een geldig gevolg is van Σ (notatie $\Sigma \vDash \psi$) indien voor elk model M en elke bedeling b geldt

als voor elke $\varphi \in \Sigma$ geldt dat $V\ M,b\ (\varphi)$= 1 dan geldt ook $V\ M,b\ \psi$= 1

Voorbeelden:
- $\forall x\ (Rx \to Px), \exists x\ Rx\ \vDash\ \exists x\ Px$
- $\forall x\ \exists y\ Rxy, \forall x\ \forall y\ (Rxy \to Ryx), \forall x\ \forall y\ \forall z\ \big((Rxy\ \wedge Ryz) \to Rxz\big) \vDash\ \forall x\ Rxx$

2.3.11 Universeel geldig

Definitie

Laat ψ een formule zijn. We zeggen dat een formule ψ universeel geldig is indien

$$\vDash \psi$$

Notatie $\vDash \Sigma$ betekent dat minstens één van de formules in Σ universeel geldig is.

Voorbeeld:

- $\vDash Ta \to \exists x\ Tx$
- $\vDash \forall x\ K(x,x)$ met K "gelijk aan"

2.3.12 Logisch equivalent

Definitie

Als voor formules φ en ψ geldt dat $\vDash \varphi\ \psi$ dat heten φ en ψ logisch equivalent.

Merk op: φ en ψ zijn logisch equivalent indien voor elk model M en elke bedeling b geldt dat $VM,b\ \varphi = V\ M,b\ \psi$.

Voorbeeld:

- $\forall x\ Rx$ en $\neg\exists x\ \neg Rx$
- $\forall x\ (Rx\ \wedge Ax)$ en $(\forall x\ Rx\ \wedge\ \forall x\ Ax)$

2.3.13 Theorie //

Definitie

De theorie van een model M is gedefinieerd als $Th\ (M)$ = $(\varphi \mathrm{I} \varphi\ een\ \mathrm{zin}\ \mathrm{en}\ M \vDash \varphi)$

2.3.14 Axiomaverzameling

Definitie

Een formuleverzameling $\boldsymbol{\Sigma}$ axiomatiseert $Th(M)$ als voor alle zinnen φ geldt:

$$\varphi \in Th(M) \Leftrightarrow \Sigma \vDash \varphi$$

We noemen $\boldsymbol{\Sigma}$ een axiomatiek of axiomaverzameling voor $Th(M)$.

Opmerking $Th(M)$ is een axiomatiek voor $Th(M)$.

Een goede axiomatiek geeft de essentiele kenmerken weer van het model.

2.3.15 Modelverzameling

Definitie

De modelverzameling van een zin φ is gedefinieerd als

$$MOD(\varphi) = \{M | M \vDash \varphi\}$$

De modelverzameling van een verzameling zinnen Σ is gedefineerd als

$$MOD(\Sigma) = \ \{M | M \vDash \varphi\ voor\ alle\ \varphi \in \Sigma\}$$

2.4 Geldig gevolg

2.4.1 Propositielogica vs predikaatlogica

We willen sematische tableaus gebruiken voor geldig gevolg aan te tonen => zoeken naar tegenvoorbeeld.
Kan niet want we ontbreken

- reductieregels voor ∀ en ∃
- construeren van een domein D voor het tegenvoorbeeld
- bijhouden interpretatiefunctie I en bedeling b

+ Leer de rest van de slides!

2.5 Afleidingen

- Bijkomende afleidingsregels ∀ en ∃
- Afleidingsregels propositielogica blijven geldig
    - (We beperken ons tot formules zonder vrije variabelen (in het boek izjn vrije variabelen toegelaten)

+ ook leren uit slides / oefeningen (61 b e, 62, 64)

2.6 Metatheorie

2.6.1 Substitutie

Eigenschap waarheidsfunctie

Laat $x_1, \dots\ x_k$ de vrije variabelen van φ zijn, b_1 en b_2 bedelingen zodat $b_1(x_i) = \ b_2(x_i)$ voor alle $i = 1, \dots, k$ en een model M.
Dan geldt $V_{M,b_1}(\varphi) = V_{M,b_2}(\varphi)$.

Eigenschap 1 substitutie:

Voor alle termen t en t' en een variabele x geldt

$$V_{M,b}([t/x]t') = V_{M,b[x\mapsto V_{M,b}(t)]}(t')$$

voor alle M en b.

Bewijs:

Bewijs

$$V_{M,b}([t/x]t') = V_{M,b[x\mapsto V_{M,b}(t)]}(t')$$

We bewijzen dit met inductie naar de opbouw van t'.

1. Stel $t' = x$, dan
   - $V_{M,b}([t/x]t') = V_{M,b}([t/x]x) = V_{M,b}(t)$
   - $V_{M,b[x\mapsto V_{M,b}(t)]}(t') = V_{M,b[x\mapsto V_{M,b}(t)]}(x) = V_{M,b}(t)$
2. Stel $t' = y \neq x$ met y een variabele of constante, dan
   - $V_{M,b}([t/x]t') = V_{M,b}([t/x]y) = V_{M,b}(y)$
   - $V_{M,b[x\mapsto V_{M,b}(t)]}(t') = V_{M,b[x\mapsto V_{M,b}(t)]}(y) = V_{M,b}(y)$

$$V_{M,b}([t/x]t') = V_{M,b[x\mapsto V_{M,b}(t)]}(t')$$

3. Inductiehypothese: $V_{M,b}([t/x]t_i) = V_{M,b[x\mapsto V_{M,b}(t)]}(t_i)$ voor alle M en b, dan
   - $V_{M,b}([t/x]f(t_1, \dots, t_k)) = V_{M,b}(f([t/x]t_1, \dots, [t/x]t_k)) = I(f)\left(V_{M,b}([t/x]t_1), \dots, V_{M,b}([t/x]t_k)\right) = \mathrm{I(f)}(V_{M,b[x\mapsto V_{M,b}(t)]}(t_1), \dots, V_{M,b[x\mapsto V_{M,b}(t)]}(t_k))$
   - $V_{M,b[x\mapsto V_{M,b}(t)]}(f(t_1, \dots, t_k)) = I(f)\left(V_{M,b[x\mapsto V_{M,b}(t)]}(t_1), \dots, V_{M,b[x\mapsto V_{M,b}(t)]}(t_k)\right)$

Eigenschap 2 substitutie:

Voor een formule φ en variabele x zodat t vrij is voor x in φ geldt in elke model M en voor elke bedeling b dat

$$V_{M,b}([t/x]\varphi) = V_{M,b[x \mapsto V_{M,b}(t)]}(\varphi)$$

Bewijs:

Bewijs

$$V_{M,b}([t/x]\varphi) = V_{M,b[x \mapsto V_{M,b}(t)]}(\varphi)$$

We bewijzen dit met inductie naar de opbouw van φ.

1. Inductiehypothese: $V_{M,b}([t/x]t_i) = V_{M,b[x \mapsto V_{M,b}(t)]}(t_i)$ voor alle M en b. Stel $\varphi = P(t_1, \dots, t_k)$, dan
   - $V_{M,b}([t/x]\varphi) = V_{M,b}([t/x]P(t_1, \dots, t_k)) = V_{M,b}(P([t/x]t_1, \dots, [t/x]t_k))$
     $= I(P)(V_{M,b}([t/x](t_1)), \dots, V_{M,b}([t/x](t_k)))$
     $= \mathrm{I(P)}(V_{M,b[x \mapsto V_{M,b}(t)]}(t_1), \dots, V_{M,b[x \mapsto V_{M,b}(t)]}(t_k))$
   - $V_{M,b[x \mapsto V_{M,b}(t)]}(\varphi) = V_{M,b[x \mapsto V_{M,b}(t)]}(P(t_1, \dots, t_k))$
     $= \mathrm{I(P)}(V_{M,b[x \mapsto V_{M,b}(t)]}(t_1), \dots, V_{M,b[x \mapsto V_{M,b}(t)]}(t_k))$
2. Te bewijzen voor alle connectieven. Hier ter illustratie bewijzen we voor negatie $\varphi = \neg\psi$ met inductiehypothese $V_{M,b}([t/x]\psi) = V_{M,b[x \mapsto V_{M,b}(t)]}(\psi)$ voor alle M en b.

   Dan $V_{M,b}([t/x]\varphi) = 1$ desda $V_{M,b}([t/x]\psi) = 0$ desda $V_{M,b[x \mapsto V_{M,b}(t)]}(\psi) = 0$ desda $V_{M,b[x \mapsto V_{M,b}(t)]}(\varphi) = 1$.
3. Te bewijzen voor beide kwantoren, ter illustratie hier $\varphi = \forall z\psi$ met inductiehypothese $V_{M,b}([t/x]\psi) = V_{M,b[x \mapsto V_{M,b}(t)]}(\psi)$ voor alle M en b. We bekijken twee gevallen:
   - x is niet vrij in φ en dus $[t/x]\varphi = \varphi$ (definitie: enkel de vrije variabelen) zodat $V_{M,b}([t/x]\varphi) = V_{M,b}(\varphi)$. Omdat x niet vrij is, geldt volgens een vorige bewering dat $V_{M,b[x \mapsto V_{M,b}(t)]}(\varphi) = V_{M,b}(\varphi)$.
   - x is vrij in φ en dus $[t/x]\varphi = [t/x]\forall z\psi = \forall z\, [t/x]\psi$ zodat $V_{M,b}([t/x]\varphi) = V_{M,b}(\forall z\, [t/x]\psi)$. Dit betekent $V_{M,b}([t/x]\varphi) = 1$ desda voor alle $d \in D$: $V_{M,b[z \mapsto d]}([t/x]\psi) = 1$.

We onderzoeken $V_{M,b[z \mapsto d]}([t/x]\psi)$. Uit de inductiehypothese (neem $b[z \mapsto d]$) weten we dat in het bijzonder $V_{M,b[z \mapsto d]}([t/x]\psi) = V_{M,b[z \mapsto d][x \mapsto V_{M,b[z \mapsto d]}(t)]}(\psi)$. Gegeven is dat t vrij is voor x in φ, dus z komt zeker niet voor in t, dus $V_{M,b[z \mapsto d]}(t) = V_{M,b}(t)$. Extra aanname is dat x vrij is in φ dus $x \neq z$ (anders was x gebonden). Er volgt $b[z \mapsto d][x \mapsto V_{M,b}(t)] = b[x \mapsto V_{M,b}(t)]\,[z \mapsto d]$. Samengevat: $V_{M,b[z \mapsto d]}([t/x]\psi) = V_{M,b[x \mapsto V_{M,b}(t)][z \mapsto d]}(\psi)$.

We hernemen $V_{M,b}([t/x]\varphi) = 1$ desda voor alle $d \in D$: $V_{M,b[x \mapsto V_{M,b}(t)][z \mapsto d]}(\psi) = 1$. Dit is per definitie desda $V_{M,b[x \mapsto V_{M,b}(t)]}(\forall z\psi) = 1$. Dus desda $V_{M,b[x \mapsto V_{M,b}(t)]}(\varphi) = 1$

2.6.2 Lemma

Equivalenties ∀, ∃, ¬: lemma

Als φ en ψ formules zijn en x is een variabele, dan zijn volgende formules logisch equivalent:

- $\forall x \neg\varphi$ en $\neg\exists x\varphi$
- $\exists x \neg\varphi$ en $\neg\forall x\varphi$

Lemma bewijs

Bewijs

We tonen als illustratie dat $\forall x \neg\varphi$ en $\neg\exists x\varphi$ logisch equivalent zijn

$V_{M,b}\ (\forall x\ \neg\varphi) = 1$

desda voor alle $d \in D$: $V_{M,b[x \mapsto d]}\ (\neg\varphi) = 1$

desda voor alle $d \in D$: $V_{M,b[x \mapsto d]}\ (\varphi) = 0$

desda er bestaat geen $d \in D$: $V_{M,b[x \mapsto d]}\ (\varphi) = 1$

desda $V_{M,b}\ (\exists x\ \varphi) = 0$

desda $V_{M,b}\ (\neg\exists x\ \varphi) = 1$

Equivalenties ∀, ∃, ∧,∨: lemma

Als φ en ψ formules zijn en x is een variabele is die niet vrij voorkomt in $\boldsymbol{\psi}$, dan zijn volgende formules logisch equivalent:

- $(\exists x\varphi) \wedge \psi$ en $\exists x\ (\varphi \wedge \psi)$
- $(\forall x\varphi) \wedge \psi$ en $\forall x\ (\varphi \wedge \psi)$
- $(\exists x\varphi) \vee \psi$ en $\exists x\ (\varphi \vee \psi)$
- $(\forall x\varphi) \vee \psi$ en $\forall x\ (\varphi \vee \psi)$
- $\psi \wedge (\exists x\varphi)$ en $\exists x\ (\psi \wedge \varphi)$
- $\psi \wedge (\forall x\varphi)$ en $\forall x\ (\psi \wedge \varphi)$
- $\psi \vee (\exists x\varphi)$ en $\exists x\ (\psi \vee \varphi)$
- $\psi \vee (\forall x\varphi)$ en $\forall x\ (\psi \vee \varphi)$

Bewijs

We bewijzen $(\exists x\varphi) \wedge \psi$ en $\exists x\ (\varphi \wedge \psi)$ als illustratie.

$V_{M,b}\left((\exists x\varphi) \wedge \psi\right) = 1$

desda $V_{M,b}(\exists x\varphi) = 1$ en $V_{M,b}\ (\psi) = 1$

desda er bestaat $d \in D$: $V_{M,b[x\mapsto d]}(\varphi) = 1$ en $V_{M,b}\ (\psi) = 1$

desda er bestaat $d \in D$: $V_{M,b[x\mapsto d]}(\varphi) = 1$ en $V_{M,b[x\mapsto d]}\ (\psi) = 1$ (x niet vrij in ψ)

desda er bestaat $d \in D$: $V_{M,b[x\mapsto d]}(\varphi \wedge \psi) = 1$

desda $V_{M,b}\ (\exists x\ (\varphi \wedge \psi)) = 1$

Equivalenties ∀, ∃, ->: lemma

Als φ en ψ formules zijn en x is een variabele is die niet vrij voorkomt in $\boldsymbol{\psi}$, dan zijn volgende formules logisch equivalent:

- $(\forall x\varphi) \rightarrow \psi$ en $\exists x\ (\varphi \rightarrow \psi)$
- $(\exists x\varphi) \rightarrow \psi$ en $\forall x\ (\varphi \rightarrow \psi)$
- $\psi \rightarrow (\forall x\varphi)$en $\forall x\ (\psi \rightarrow \varphi)$
- $\psi \rightarrow (\exists x\varphi)$en $\exists x\ (\psi \rightarrow \varphi)$

Bewijs

We bewijzen $(\forall x\varphi) \rightarrow \psi$ en $\exists x\ (\varphi \rightarrow \psi)$ als illustratie.

$(\forall x\varphi) \rightarrow \psi \equiv \neg\ (\forall x\varphi) \vee \psi \equiv \exists x(\neg\varphi) \vee \psi \equiv \exists x\ (\neg\varphi \vee \psi) \equiv \exists x(\varphi \rightarrow \psi)$

2.6.3 Prenexvorm

Prenexstelling:

Voor elke formule φ bestaat er steeds een logisch equivalente formule in prenexvorm.

Waarbij prenexvorm betekent dat de formule van de vorm $Q_1x_1 \dots Q_nx_n\psi$ is met kwantoren $Q_1, \dots, Q_n$ zodat er geen kwantoren meer voor komen in ψ.

Bewijs

Via inductie naar φ (combinatie van de lemma's)

2.6.4 Fragmenten van predikaatlogica ????

2.6.5 Adequaatheid van tableaus ????

2.6.6 Volledigheidsstelling ????

3 Lambda calculus

Lambda calculus vormt de basis van functionele programeertalen zoals lisp en scheme

Notatie:
Voorbeeld: $f(x) = 2 + x = +(2)(x)$

$$\lambda x.\big((+)2\big)x$$

Vorm: λ <parameter>.<voorschrift>
Toepassing functie: (functie)argument

Functie heeft geen naam + Heeft maar 1 input nodig

3.1.1 Basis begrippen

Abstractie: aanmaken van een functievoorschrift
Applicatie: aanroepen van een functievoorschrift

3.1.2 Lambda expressies (syntax)

Definitie

Zij V een verzameling variabelen. De verzameling van λ-expressies Λ wordt alsvolgt gedefinieerd:

1. $V \subseteq \Lambda$
2. Als $M \in \Lambda$ en $\mathrm{N} \in \Lambda$ dan is $(M)N \in \Lambda$. (applicatie)
3. Als $x \in V$ en $M \in \Lambda$ dan is $\lambda x.M \in \Lambda$. (abstractie)
4. Niets anders zit in Λ.

Intuitie: slide 14

Opmerkingen: // Slide 15

Voorbeeld: // slide 16

3.1.3 Currying

Een methode om meerdere inputs te hebben bij lambda calculus ///

Een functie kan een andere functie hebben als output bv : f(x,y) = x + y => λx. λy .((+) y) x)

3.1.4 Binding en bereik

Voor een λ-expressie $\lambda x.M$ zegt men dat

- λx een binding is van x in M
- het bereik van de binding M is: alle (nog niet gebonden) voorkomens van x in $\lambda x.M$ zijn gebonden

In een λ-expressie heten alle voorkomens van variabelen die niet gebonden zijn vrij

3.1.5 Vrije variabelen

De vrije variabelen van een λ-expressie zijn alsvolgt gedefinieerd:

- $\forall x \in V: VV(x) = \{x\}$
- $\forall M, N \in \Lambda: VV(M(N)) = VV(M) \cup VV(N)$
- $\forall x \in V, \forall M \in \Lambda: VV(\lambda x. M) = VV(M) \setminus \{x\}$

Notatie: $VV(M)$ van vrije variabelen van M.

3.1.6 Gesloten lambda-expressie

Een λ-expressie zonder vrije variabelen heet gesloten of een combinator. De verzameling van combinatoren wordt genoteerd als Λ_0.

Voorbeelden:

3.1.7 Substitutie: definitie

Definitie

Beschouw λ-expressies P en M en $x \in V$. De substitutie $[P/x]M$ van P voor x in M wordt alsvolgt gedefinieerd:

(S1) $[P/x]x = \mathrm{P}$

(S2) $[P/x]y = y$ als $y \in V \setminus \{x\}$

(S3) $[P/x](F)Q = ([P/x]F)[P/x]Q$

(S4) $[P/x]\lambda x. M = \lambda x. M$

(S5) $[P/x]\lambda y. M = \lambda y. [P/x]M$ als $y \neq x$ en $y \notin VV(P)$

(S6) $[P/x]\lambda y. M = \lambda z. [P/x][z/y]M$ als $y \neq x$ en $z \notin VV(P)$ en $y \in VV(P)$

Intuiettief:

Voorbeelden:

3.2 Rekenen met lambda experssies

3.2.1 Beta-gelijkheid

Definitie

De relatie $=_\beta \subseteq \Lambda \times \Lambda$ wordt gedefinieerd door de axioma's:

$(\beta) (\lambda x. M)P =_\beta [P/x]\mathrm{M}$

(α) $\lambda x. M =_\beta \lambda z. [z/x]M$ als $z \notin VV(M)$

(reflexief) $M =_\beta M$

(symmetrisch) $M =_\beta N$ dan $N =_\beta M$

(transitief) $M =_\beta N$ en $\mathrm{N} =_\beta L$, dan $M =_\beta L$

(congruent) $M =_\beta M'$ en $\mathrm{P} =_\beta P'$, dan $(M)P =_\beta (M')P'$

(congruent) $M =_\beta M'$ dan $\lambda x. M =_\beta \lambda x. M'$

Intuitie:

3.2.2 Church getallen

Lambda calculus heeft geen constanten. Toch kunnen we natuurlijke getallen en rekenkundige functies voorstellen.

Definitie

De zogenaamde Church getallen c_n $(n \in \mathbb{N})$ worden gedefinieerd door

$$c_n \equiv \lambda f.\lambda x.(f)^n x$$

Intuïtief: gegeven een functie f en een parameter x, passen we f n keer toe op x.

3.2.3 Lemma ("+")

Beschouw λ-expressies F en M.Voor alle $m, n \in \mathbb{N}$ geldt

$$(F)^{n+m}M \equiv (F)^n(F)^m M$$

Lemma bewijs

Bewijs

We bewijzen dit via inductie op n.
Als $n = 0$. Dan $(F)^{n+m}M \equiv (F)^m M \equiv_{DEF} (F)^0(F)^m M \equiv (F)^n(F)^m M$

Inductiehypothese: bewering geldt voor $\mathrm{n} = k$ $(F)^{k+m}M \equiv (F)^k(F)^m M$.
We tonen voor $\mathrm{n} = k + 1$
$(F)^{(k+1)+m}M \equiv (F)^{1+(k+m)}M \equiv_{DEF} (F)(F)^{k+m}M \equiv_{IH} (F)(F)^k(F)^m M$
$\equiv_{DEF} (F)^{1+k}(F)^m M \equiv (F)^{k+1}(F)^m M$

3.2.4 Lambda-definieerbaar

Definitie

Een numerieke functie $f: \mathbb{N}^p \rightarrow \mathbb{N}$ is λ-definieerbaar als er een combinator F bestaat zodat

$$((((F)c_{n_1})c_{n_2}) \dots)c_{n_p} =_\beta c_{f(n_1,\dots,n_p)}$$

voor $n_1, \dots, n_p \in \mathbb{N}$
Intuïtief: er bestaat een combinator waarvan het effect op de Church getallen hetzelfde is als het effect op de natuurlijke getallen

Voorbeeld:

3.2.5 Optelling

$$plus \equiv \lambda n.\lambda m.\lambda f.\lambda x.((n)f)((m)f)x$$

Dan geldt voor alle $m, n \in \mathbb{N}$: $((plus)c_n)c_m =_\beta c_{n+m}$

Lemma bewijs

Bewijs

$((plus)c_n)c_m \equiv ((\lambda n.\lambda m.\lambda f.\lambda x.((n)f)((m)f)x)c_n)c_m$	(def)
$=_\beta (\lambda m.\lambda f.\lambda x.((c_n)f)((m)f)x)c_m$	(β)
$\equiv (\lambda m.\lambda f.\lambda x.((\lambda f.\lambda x.(f)^n x)f)((m)f)x)c_m$	(def)
$=_\beta (\lambda m.\lambda f.\lambda x.(\lambda x.(f)^n x)((m)f)x)c_m$	(β)
$=_\beta (\lambda m.\lambda f.\lambda x.(f)^n((m)f)x)c_m$	(β)
$=_\beta \lambda f.\lambda x.(f)^n((c_m)f)x$	(β)
$\equiv \lambda f.\lambda x.(f)^n((\lambda f.\lambda x.(f)^m x)f)x$	(def)
$=_\beta \lambda f.\lambda x.(f)^n(\lambda x.(f)^m x)x$	(β)
$=_\beta \lambda f.\lambda x.(f)^n(f)^m x$	(β)
$\equiv \lambda f.\lambda x.(f)^{n+m} x$	(lemma)
$\equiv c_{n+m}$	

3.2.6 Lemma (“*”)

Voor alle $m, n \in \mathbb{N}$ geldt

$$((c_n)f)^m y =_\beta (f)^{n\times m} y$$

Intuïtief: een soort kopieerfunctie: m keer na elkaar $((c_n)f)$ toepassen

Bewijs

We bewijzen dit via inductie op m.

Als $m = 0$.

$((c_n)f)^m y \equiv ((c_n)f)^0 y$	
$\equiv y$	(def)
$\equiv (f)^0 y$	(def)
$\equiv (f)^{n\times 0} y$	
$\equiv (f)^{n\times m} y$	

Inductiehypothese: bewering geldt voor $m = k$: $((c_n)f)^k y =_\beta (f)^{n\times k} y$

We tonen voor $m = k + 1$

$((c_n)f)^{k+1} y \equiv ((c_n)f)((c_n)f)^k y$	(def)
$=_\beta ((c_n)f)(f)^{n\times k} y$	(IH)
$\equiv ((\lambda f.\lambda x.(f)^n x)f)(f)^{n\times k} y$	(def)
$=_\beta (\lambda x.(f)^n x)(f)^{n\times k} y$	(β)
$=_\beta (f)^n(f)^{n\times k} y$	(β)
$\equiv (f)^{n+n\times k} y$	(lemma)
$\equiv (f)^{n\times(1+k)} y$	
$\equiv (f)^{n\times(k+1)} y$	

Vermenigvuldiging

Bewijs

$$
\begin{aligned}
((times)c_n)c_m &\equiv ((\lambda n.\lambda m.\lambda f.(n)(m)f)c_n)c_m && \text{(def)}\\
&=_\beta (\lambda m.\lambda f.(c_n)(m)f)c_m && (\beta)\\
&\equiv (\lambda m.\lambda f.(\lambda f.\lambda x.(f)^n x)(m)f)c_m && \text{(def)}\\
&=_\beta \left(\lambda m.\lambda f.\lambda x.((m)f)^n x\right)c_m && (\beta)\\
&=_\beta \lambda f.\lambda x.((c_m)f)^n x && (\beta)\\
&=_\beta \lambda f.\lambda x.(f)^{m\times n} x && \text{(lemma)}\\
&\equiv c_{m\times n} && \text{(def)}\\
&\equiv c_{n\times m}
\end{aligned}
$$

3.2.7 True, false, if…

Bekijk slides voor meer info 36 - 51

$$
\begin{aligned}
&true \equiv \lambda t.\lambda f.t\\
&false \equiv \lambda t.\lambda f.f\\
&if \equiv \lambda c.\lambda d.\lambda e.((c)d)e\\
&iszero \equiv \lambda n.((n)\lambda x.false)true\\
&cons \equiv \lambda a.\lambda d.\lambda z.((z)a)d\\
&car \equiv \lambda a.\lambda d.a\\
&cdr \equiv \lambda a.\lambda d.d
\end{aligned}
$$

3.3 Fixpunten

3.3.1 Fixpunten definitie

Definitie

Voor een functie $f: D \to D$ heet $x \in D$ een fixpunt van f desda

$$f(x) = x$$

$X \in \Lambda$ is een fixpunt van $F \in \Lambda$ desda

$$(F)X =_\beta X$$

3.3.2 Fixpunten stelling

Voor elke $F \in \Lambda$ bestaat er een $X \in \Lambda$ zodat

$$(F)X =_\beta X.$$

Voor

$$Y \equiv \lambda f.(\lambda x.(f)(x)x)\lambda x.(f)(x)x$$

geldt voor alle $F \in \Lambda$

$$(F)(Y)F =_\beta (Y)F.$$

Y wordt de Y-combinator genoemd.

Bewijs

Definieer $W \equiv \lambda x.(F)(x)x$ en $X \equiv (W)W$. Dan
$X \equiv (W)W$

$$\equiv (\lambda x.(F)(x)x)W$$
$$=_\beta (F)(W)W$$
$$\equiv (F)X$$

Dus X is een fixpunt van F.

$(Y)F \equiv (\lambda f.(\lambda x.(f)(x)x)\lambda x.(f)(x)x\,)F$

$$=_\beta (\lambda x.(F)(x)x)\lambda x.(F)(x)x$$
$$\equiv (W)W$$

Er volgt dat $(Y)F$ een fixpunt is van F want $(W)W$ is een fixpunt:

$$(F)(Y)F =_\beta (F)(W)W =_\beta (W)W =_\beta (Y)F$$

3.3.3 Recursie

Recursieve functie faculteit:

$$fac(n) = \begin{cases} 1, & n = 0 \\ n \cdot fac(n-1), & n \geq 0 \end{cases}$$

$(((if)\ cond)\ A)\ B$

Letterlijke vertaling:

$$fac \equiv \lambda n.(((if)(iszero)n)c_1)((times)n)(fac)(pred)n$$

circulaire definitie! -> soort oneindig lang word, mag niet

oplossing: we vervangen $\equiv$ door $=_\beta$

Samenvatting:

$Y \equiv \lambda f.(\lambda x.(f)(x)x)\lambda x.(f)(x)x$

$FAC \equiv \lambda f.\lambda n.\left(((if)(iszero)n)c_1\right)((times)n)(f)(pred)n$

$fac \equiv (Y)FAC$

$W_{FAC} \equiv \lambda x.(FAC)(x)x$

Toon als oefening: $(fac)c_0 =_\beta c_1$, $(fac)c_1 =_\beta c_1$

Extra: (+ slide 10 oef)

3.3.4 Gevolg

Beschouw een $F \in \Lambda$ van de vorm

$$\lambda f.M$$

en fixpunt

$$X_F \equiv (Y)F.$$

Dan geldt

$$X_F =_\beta [X_F/f]M$$

Bewijs

X_F is een fixpunt van F, dus per definitie: $(F)X_F =_\beta X_F$.

Uit het β axioma volgt:

$$(F)X_F \equiv (\lambda f.M)X_F =_\beta [X_F/f]\mathrm{M}.$$

Samengevat:

$$X_F =_\beta [X_F/f]\mathrm{M}$$