

Les Formele Logica 02 - 04 - 2004

Benjamin De Leeuw

Departement Zuivere Wiskunde en Computer Algebra
Galglaan 2, B-9000 Gent
bdeleeuw@cage.ugent.be
Gent University, Belgium

April 1, 2004

1 Doelstelling

1. Taal van de Predikaatlogica
2. Tarski's World
3. Oefeningen op Tarski's World
4. Tarski's World als Spel

2 Taal van de Predikaatlogica

2.1 Signatuur, Functies en Relaties

- Een **individu** of object (letterlijk : datgene waarover men spreekt) is van een bepaald type of **soort**.
- We verzamelen alle soorten waarover we predikaten willen definiëren in een soortenverzameling.
- Strengen (sequenties) van soorten uit de soortenverzameling vormen de signatuur van **functies** en **predikaten** en worden **woorden** genoemd.
- De verzameling van alle woorden (van willekeurige lengte) zonder het lege woord wordt S^+ genoemd.
- De verzameling van alle woorden (van willekeurige lengte) met het lege woord wordt S^* genoemd.
- Objecten van soort s worden gegroepeerd in verzamelingen A_s .
- De collectie van de verzamelingen A_s wordt met A aangeduid en is een **s-soortige verzameling**.

In symbolen:

elk object o is van een soort

$$S := \{s \mid s \text{ is een soort}\}$$

$$S_n := \{w \mid w = s_1 s_2 \dots s_n \wedge s_i \in S\}$$

$$S^+ := \bigcup_{n=1}^{\infty} S_n$$

$$S^* := S^+ \cup \{\phi\} \quad (|\phi| = 0)$$

$$A_s := \{o \mid o \text{ is object van soort } s\} \quad (s \in S)$$

$$A := \{A_s \mid s \in S\}$$

- operationele signatuur: $\Omega := \{\Omega_{w,s} \mid \langle w, s \rangle \in S^* \times S\}$
- relationele signatuur: $\Pi := \{\Pi_w \mid w \in S^*\}$
- signatuur van een *theorie*: $\Sigma := \langle \Omega, \Pi \rangle$

Het alfabet van **PRED** _{Σ} is:

1. $\{\neg, \wedge, \vee, =, (,)\}$
2. $\Sigma := \langle \Omega, \Pi \rangle$
3. $X := \{X_s \mid s \in S\}$

- De operationele signatuur definiëert alle mogelijke functievoorschriften en return-waarden over de beschouwde types S (denk aan java).
- De relationele signatuur bepaalt alle mogelijke relaties die te definiëren zijn over de beschouwde types S (denk aan prolog).
- Een relatie kan opgevat worden als een functie die een waarde van het type boolean teruggeeft.
- Een functie kan opgevat worden als een relatie, over de attributtypes geconcateneerd met het return-waardetype, die waar is indien de return-waarde het juiste antwoord is van de functie.

2.2 Syntaxis van de Predikaatlogica

$$\langle term \rangle ::= c \mid v \mid f_1 \langle term \rangle \mid \dots \mid f_n \langle term \rangle \dots \langle term \rangle$$

met

$$\begin{array}{ll} c \in \Omega_{\phi,s} & s \in S \\ v \in X & X := \{X_s \mid s \in S\} \\ f_i \in \Omega_{w,s} & s \in S, \mid w \mid = i \end{array}$$

$$\langle atfo \rangle ::= p \mid q_1 \langle term \rangle \mid \dots \mid q_n \langle term \rangle \dots \langle term \rangle \mid \langle term \rangle = \langle term \rangle$$

met

$$\begin{array}{ll} p \in \Pi_{\phi} & \\ q_i \in \Pi_w & \mid w \mid = i \\ \langle term \rangle = \langle term \rangle & \text{enkel over termen van dezelfde soort} \end{array}$$

$$\langle formule \rangle ::= \langle atfo \rangle \mid \neg \langle formule \rangle \mid \langle formule \rangle \wedge \langle formule \rangle \mid \forall v \langle formule \rangle \mid (\langle formule \rangle)$$

met

$$v \in X \quad X := \{X_s \mid s \in S\}$$

2.3 Vrije en gebonden Variabelen, Substitutie, Varianten

- Een variabele is vrij indien ze niet gebonden is. Een variabele $v \in X$ is gebonden indien ze in het bereik van de kwantor $\forall v$ valt. Het bereik van de $\forall$ -kwantor wordt bepaald door de syntaxis uit ?? ($\langle formule \rangle$ is het bereik van de kwantor). Voorbeeld: $\forall v f_1(v) = c \wedge q_1(v)$ en $\forall v (f_1(v) = c \wedge q_1(v))$.
- Een variabele die vrij is, kan door een willekeurige andere $\langle term \rangle$ vervangen worden, notatie $[t/v]\varphi$, met t een $\langle term \rangle$. $[t/v]\varphi$ wordt ook een instantie genoemd van φ . Voorbeeld: $[f_2(v, f_1(c))/v]f_1(v) \equiv f_1(f_2(v, f_1(c)))$.
- Door substitutie kan in de plaats van een vrije variabele, een gebonden gesubstitueerd voorkomen van een andere variabele ontstaan (in het bereik van een kwantor). Dit is een ongewenst resultaat. Om dit te voorkomen wordt een herbenoeming doorgevoerd van de variabelen in kwestie. Men noemt dit een alfabetische variant. Voorbeeld: $[f_2(\mathbf{x}, f_1(c))/v]\forall \mathbf{x}(f_1(\mathbf{x}) \wedge q_1(v)) \equiv \forall \mathbf{u}(f_1(\mathbf{u}) \wedge q_1(f_2(x, f_1(c)))) \neq \forall \mathbf{x}(f_1(\mathbf{x}) \wedge q_1(f_2(\mathbf{x}, f_1(c))))$.

3 Tarski's World

3.1 Inleiding

- Tarski's World is een programma dat op een intuïtieve manier de begrippen van de eerste orde logica aanbrengt. Men kan predikaatformules opstellen en verifiëren over werelden, bewoond door geometrische figuren.
- Tarski's World in de PC-zalen: 'F:/ahogew/Tarski/Tarski.exe'. Alle voorbeeldwerelden en -zinnen staan daar ook.
- Tarski's World heeft vier vensters: het wereldvenster, het zinnenvenster, het inspectorvenster en het keyboardvenster.
- Zowel werelden (worlds,*.WLD) als zinnen (sentences,*.SEN) kunnen geladen en opgeslagen worden.

3.2 Syntaxis van formules in Tarski's World

Het alfabet van **TARSKI** is:

1. $S := \{m\}$
2. $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists, =, \neq, (,)\}$
3. $\Sigma := \langle \Omega, \Pi \rangle$
4. $X := \{u, v, w, x, y, z\}$

met

$$\begin{aligned}\alpha \vee \beta &:= \neg(\neg\alpha \wedge \neg\beta) \\ \alpha \rightarrow \beta &:= (\neg\alpha \vee \beta) \\ \alpha \leftrightarrow \beta &:= ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)) \\ \exists v\alpha &:= \neg\forall v\neg\alpha \\ t_1 \neq t_2 &:= \neg(t_1 = t_2) \\ \Omega_{\phi,m} &:= \{a, b, c, d, e, f\} \\ \Omega_{w,m} &:= \phi \quad (\forall w \mid w \mid > 0) \\ \Pi_m &:= \{Tet, Cube, Dodec, Small, Medium, Large\} \\ \Pi_{mm} &:= \{Smaller, Larger, LeftOf, RightOf, BackOf, FrontOf\} \\ \Pi_{mmm} &:= \{Between\}\end{aligned}$$

Een uitdrukking (expression) in het Tarski's World zinnenvenster kan op drie manieren geëvalueerd worden:

1. **wff**: well formed formula, is een uitdrukking die correct geformuleerd is in **TARSKI**.
2. **sentence**: een wff is een zin, indien er geen vrije variabelen in voorkomen.
3. **verify**: een zin kan geëvalueerd worden naar waarheid in de huidige wereld (world).

3.3 Model en Valuatie

Een model is een tupel $\mathcal{M} := \langle D, \mathcal{I} \rangle$. Een wereld in Tarski's World is zo een model, met de volgende betekenis:

$D := D_m$	$\{(large, dodec, [5, 2], c), (medium, tet, [3, 3], null), \dots\}$ (de geometrische objecten in de wereld)	(max 12)
$\mathcal{I}(\Omega_{\phi, m})$	$\{(c, (large, dodec, [5, 2], c)), \dots\}$ (gelabelde geometrische objecten in de wereld)	
$\mathcal{I}(\Omega_{w, m})$	$\phi \quad (\forall w \mid w \mid > 0)$ (er zijn geen echte functies in Tarski's World)	
$\mathcal{I}(\Pi_m)$	$\mathcal{I}(Dodec(v)) := (v.dodec) ? 1 : 0$ (objecten die voldoen aan de semantiek van het unair predikaat)	
$\mathcal{I}(\Pi_{mm})$	$\mathcal{I}(FrontOf(v, w)) := (w.y > v.y) ? 1 : 0$ (objecten die voldoen aan de semantiek van het binair predikaat)	
$\mathcal{I}(\Pi_{mmm})$	$\mathcal{I}(Between(u, v, w)) := (\mathcal{I}(FrontOf(v, u)) \wedge \dots) ? 1 : 0$ (extra opgave : vervolledig '...' in functie van binaire predikaten)	

Een valuatie $\mathcal{V}$ kent een waarheidswaarde toe aan predikaatformules ($\mathcal{V} : \mathbf{TARSKI} \mapsto \{0, 1\}$). Ze wordt afgeleid uit de interpretatie, door de variabelen ($x \in X$) te vervangen door de effectieve objecten uit de wereld. Hierdoor kunnen de interpretaties van de predikaten effectieve waarheidswaarden berekenen. In Tarski's World kan een wff enkel geverifieerd (verify) worden indien het een zin (sentence) is (geen vrije variabelen). Voorbeelden:

$$\mathcal{V}(Tet(a)) := \mathcal{I}(Tet(\mathcal{I}(a)))$$

$$\mathcal{V}(\forall v Tet(v)) := \mathcal{I}(Tet((medium, tet, [3, 3], null))) \wedge \mathcal{I}(Tet((large, dodec, [4, 3], a))) \wedge \dots$$

(... voor elk object ($d \in D_s$) in de wereld)

$$\mathcal{V}(\exists v LeftOf(v, a)) := \mathcal{I}(LeftOf((medium, tet, [3, 3], null), \mathcal{I}(a))) \vee$$

$$\mathcal{I}(LeftOf((large, dodec, [4, 3], a), \mathcal{I}(a))) \vee \dots$$

(... voor een object ($d \in D_s$) in de wereld)

We weten dat:

$$\mathcal{M} \models \alpha \quad \text{asa} \quad \mathcal{V}_{\mathcal{M}}(\alpha) \equiv 1$$

we weten nu ook dat werelden in Tarski's World modellen zijn en de valuatie $\mathcal{V}$ bepalen. Model voor een lijst van formules en logisch gevolg hebben dezelfde definitie als in de propositielogica.

3.4 Overige mechanismen

- De uitdrukking die geëvalueerd wordt, is de huidige uitdrukking (met de cursor op) in het zinnenvenster.

- Om andere uitdrukkingen te verifiëren, gebruikt men de pijltjestoesten waarmee men de cursor verplaatst naar de relevante uitdrukking.
- Om meerdere uitdrukkingen in het zinnenvenster op te stellen gebruikt men **Add After** in de **Edit** menu, de enter-toets in het keyboardvenster of shift-enter op het keyboard. Wanneer men met de muis voor een zin klikt, wanneer de cursor veranderd is in een ronde pijl, wordt ook een leeg vak toegevoegd.
- Om een uitdrukking te verwijderen en de overige uitdrukkingen te henummeren, gebruikt men **Delete Sent** in de **Edit** menu.
- Commentaar na uitdrukkingen wordt aangegeven met ‘;’ of ‘ ” ’.
- Werelden en uitdrukkingen kunnen geopend en opgeslagen worden met de menuopties **Open** en **Save(As)** in de **File** menu.
- Het programma wordt verlaten met **Exit** in de **File** menu.
- Sluit een venster door het te selecteren en **Close** uit de **File** menu aan te klikken. Men kan ook **Close All** selecteren om alle vensters te sluiten.
- Het keyboardvenster en inspectorvenster kunnen zichtbaar gemaakt worden met de **Show Keyboard** en **Show Inspector** opties in de **Display** menu.
- Maak een nieuwe wereld aan of nieuw zinnenvenster met de **New World** en **New Sentence** opties in de **File** menu.
- Plaats blokken in het wereldvenster door een vorm te kiezen en deze te slepen naar de gewenste positie. Pas vorm, grootte en naam aan door te dubbelklikken op een blok, en het dialoogvenster te veranderen. Een blok kan ook benoemd worden door op het label in het wereldvenster te klikken.
- Verplaats blokken in het wereldvenster door ze te slepen.
- Verwijder blokken uit het wereldvenster door ze van de wereldrand te slepen.
- Om de labels (namen) van de blokken te verbergen gebruikt men de **Hide Labels** optie uit de **Display** menu. Om de labels te tonen in het wereldvenster, gebruikt men de **Show Labels** optie uit de **Display** menu.
- Omdat voor sommige werelden de zichtbaarheid van blokken in 3D niet optimaal is (Large blok voor een Small blok bvb.), kan men met de menuopties **2-D View** en **3-D View** in de **Display** menu, veranderen van zicht op de wereld.
- Met de menuopties **Rotate Left** en **Rotate Right** uit de **Display** menu kan men de wereld draaien. Let erop dat ook de predikaten over de wereld hierdoor een andere waarheidswaarde krijgen (*LeftOf*, *RightOf*, *FrontOf*, *BackOf*, ...).
- **Cut**, **Copy** en **Paste** zijn beschikbaar in de **Edit** menu.

4 Oefeningen op Tarski's World

Opgaven:

1. Experimenteer met de menu's en de vensters. Probeer de in ?? beschreven technieken uit.
2. Maak een nieuwe wereld, of laad een willekeurige wereld in, en ga volgende uitdrukkingen na:

1. $\exists x \neg \text{Cube}(x)$
2. $\neg \exists x \text{Cube}(x)$
3. $\forall x \neg \text{Cube}(x)$
4. $\neg \forall x \text{Cube}(x)$

Zijn deze uitdrukkingen wff? Zinnen? Waar? Kan je een wereld construeren waarin al deze uitdrukkingen tegelijk waar zijn?

3. Open de wereld **Peirce.wld** en de zinnen **Peirce.sen**. Ga eerst zelf (in gedachten) na welke zinnen er waar zijn. Verifiëer nadien in de wereld. Extra opgave: Schrijf de coördinaten van de blokken op en ga manueel $\mathcal{V}(\alpha)$ na door uit te schrijven in functie van het domein en de interpretatie (in **Peirce.wld**).
4. Open de zinnen **Aristotl.sen**. Construeer een wereld waarin al deze zinnen waar zijn. Let erop dat je door je wereld aan te passen, geen vroegere zinnen vals maakt.
5. Open de wereld **Maigret.wld** en de zinnen **Maigret.sen**. De blokken hebben geen labels. Geef de blokken labels zodanig dat alle zinnen waar zijn. Leidt dus uit de gegeven zinnen (allemaal waar) af welke namen de blokken moeten dragen. Geef de blokken de gevonden namen, en ga na of alle zinnen waar zijn.

5 Tarski's World als Spel

- Tarski's World is meer dan een blokkendoos die waarheidswaarden nagaat. Tarski's World is gemaakt om intuïtief te leren omgaan met de predikaatlogica. Daarom werd er ook een spel voorzien om logische formules te leren evalueren. We raden aan om wanneer men twijfelt over de waarheidswaarde van een formule, of niet verstaat waarom een formule waar of vals is, het spel te spelen met de formule in kwestie.
- Stel de waarheidswaarde (en sentence en wff) voor een complexe formule in het zinnenvenster in op positief. Klik op **Game** in het keyboardvenster. Klik op **OK**. Het spel zal nu de formule opdelen in eenvoudigere componentenformules, volgens de syntaxisregels. Bij gequantificeerde formules zal het spel vragen om één of meerdere objecten aan te duiden, die voldoen aan het predikaat bij de quantor. Versla de computer door aan te tonen dat de antwoorden in het inspectorvenster juist zijn. Door gebruik van de **Back** knop kan men een stap teruggaan, om een foute veronderstelling te herzien.
- Extra opgave: speel het spel voor opgaven 3 en 5 indien er twijfel is.