

# PREDIKAATLOGICA:



VRIJE  
UNIVERSITEIT  
BRUSSEL

# PREDIKAATLOGICA - INHOUD

Inhoud

Taal

Semantiek

Semantische Tableaus

Afleidingen

Een eenvoudige theorie

Metatheorie

# PREDIKAATLOGICA: TAAL



VRIJE  
UNIVERSITEIT  
BRUSSEL

## ► Inhoud:

Alfabet

Termen

Formules

Bereik van kwantoren

Gebonden en vrije variabelen

Gesloten en open Formules

Substitutie

Alfabetische variant

## TAAL

Predikaatlogica is een uitbreiding van de propositielogica

- Nodig om eigenschappen van individuen te beschrijven en erover te redeneren.

Voorbeeld1: Sommige landen zijn koninkrijken

Voorbeeld2: Alle koninkrijken hebben een koning

Voorbeeld3:      Aanname 1: Alle koninkrijken hebben een koning

                    Aanname 2: België is een koninkrijk

                    Conclusie: België heeft een koning

- Niet weer te geven door propositielogica
- Aanname 2 bevat eigenschap (predikaat genoemd) van een individu en er wordt over deze eigenschap een conclusie gevormd

# TAAL



De taal van predikaatlogica heeft meer uitdrukingskracht dan de taal van propositielogica en bevat daarom meer bouwstenen:

## 1. Predikaatzinnen, voorbeelden:

Brussel is een stad

Parijs is de hoofdstad van Frankrijk

Stef is kind van Kees en Ada

zeggen telkens iets over individu(en)

Predikaten worden voorgesteld via predikaatletters (hoofdletters) met vast aantal argumenten (volgorde is belangrijk!)

Individueen worden voorgesteld via constanten (kleine letters)

Voorbeelden

Alternatieve notaties

1. $S(b)$	STAD (brussel)	Sb
2. $H(p,f)$	Hoofdstad(parijs,frankrijk)	Hp <sub>f</sub>
3. $K(s,k,a)$	KIND_VAN(stef,kees,ada)	Ks <sub>ka</sub>

## 2. Kwantoren:

## uitdrukkingen van hoeveelheid

$\forall$ : universele kwantor: 'alle'

$\exists$ : existentiële kwantor: 'er is minstens één'

## Alle koninkrijken hebben een koning

**Er is minstens één** land dat een koninkrijk is.

Andere kwantoren 'de meeste', 'sommige', 'minstens 7'  
(Opgelet: maken geen deel uit van predikaatlogica)



### 3. Variabelen:

Gebruik van kwantoren introduceert de nood aan variabelen (kleine letters  $x, y, z, \dots$ )

$\forall x(Mx \rightarrow Vx)$  "Alle moeders zijn vrouwen", of "Alle mannen voetballen"

$\exists x(Fx \wedge \neg Bx)$  "Er bestaat een fiets die niet blauw is"

$\exists x((Mx \wedge Fx) \wedge Bx)$  "Er bestaat een mannelijke filosoof die een baard heeft"

$\exists x((Mx \wedge Fx) \wedge \neg Bx)$  "Er bestaat een mannelijke filosoof die een baard heeft"

$\neg \exists x((Mx \wedge Fx) \wedge Bx)$  "Er bestaat geen mannelijke filosoof die een baard heeft"

$\forall x \neg((Mx \wedge Fx) \wedge Bx)$  "Alle  $x$  (mensen) zijn niet mannelijke filosoof met baard"

$\exists x \neg((Mx \wedge Fx) \wedge Bx)$  "Er bestaat een  $x$  (mens) die geen mannelijke filosoof met baard is"







## Opgelet met volgorde

### ► Voorbeelden

$\forall$  en  $\exists$

Voor elk getal bestaat er een groter getal

Iedereen heeft een moeder (voor iedereen bestaat er een moeder)

$\exists$  en  $\forall$

Er is een verzameling die bevat is in elke verzameling

Iemand is voorouder van iedereen (eva?).

$\forall \epsilon > 0 \exists \delta > 0 \forall x, y \in \mathbb{R} (|x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon)$

$\exists \delta > 0 \forall \epsilon > 0 \forall x, y \in \mathbb{R} (|x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon)$



4. *Functions*: voorgesteld door kleine letters (f, g, ...)

$$f(x,y) = 4x+3y^2$$

$I(j)$  leeftijd van Jan

$I(j)$  nu te gebruiken in andere formules

$J(I(i), I(j))$  Ian is jonger dan Jan



## DEFINITIE ALFABET

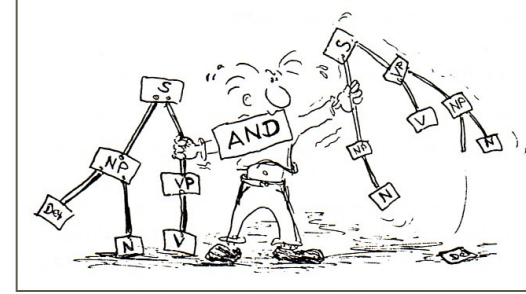
### *Definitie*

Het **alfabet** van een predikaat logische taal bestaat uit:

- ▶ Een verzameling **C** van individuele constanten:  $a, b, c, \dots, a_1, a_2, a_3, \dots$
- ▶ Een verzameling **P** van predikaatletters:  $P, Q, R, \dots, P_1, P_2, P_3, \dots$
- ▶ Een verzameling **F** van functieletters:  $f, g, h, \dots, f_1, f_2, f_3, \dots$
- ▶ De logische symbolen:  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$
- ▶ Een aantal individuele variabelen:  $u, v, w, x, y, z, x_1, x_2, x_3, \dots$
- ▶ De hulpsymbolen:  $)$  en  $($

- **Plaatsigheid** (aantal argumenten) voor functieletters en predikaatletters ook wel via een index:  $f^3abc$  of  $f^3(a, b, c)$ :  $f$  is 3-plaatsig
- Merk op: propositieletters kunnen beschouwd worden als 0-plaatsige predikaatletters.

## SYNTAXIS – DEFINITIE      TERM



### ► Termen duiden individuele objecten aan

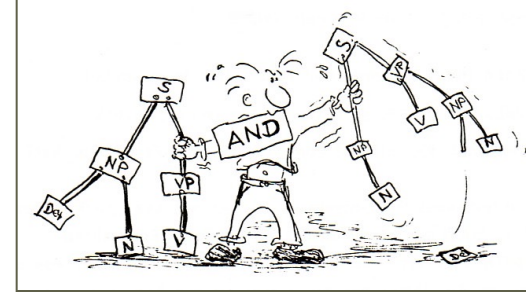
#### *Definitie*

De **termen** in de predikaatlogica zijn als volgt gedefinieerd (inductieve definitie):

- individuele variabelen en constanten zijn termen;
- als  $f$  een  $k$ -plaatsige functieletter is en  $t_1, \dots, t_k$  zijn termen, dan is  $f(t_1, \dots, t_k)$  ook een term;
- Niets anders is een term.

– Voorbeeld:  $f^3(g^2(x, h^1(y)), a, g^2(a, y))$

## OPMERKING NOTATIE



Prefix versus infixnotatie

### ► Prefixnotatie

Functiesymbool vóór de argumenten

► voorbeelden:  $f(x,y)$ ,  $+(x,y)$ ,  $\cdot(x,y)$

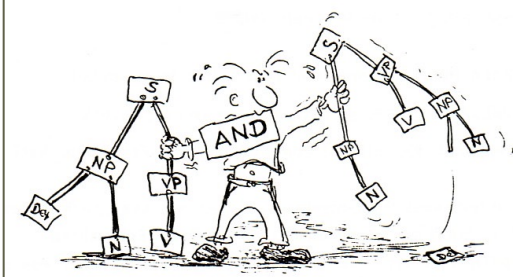
### ► Infixnotatie

Meestal bij 2-plaatsige functies

Functiesymbool tussen de argumenten

► voorbeelden:  $x + y$  en  $x \cdot y$

Meestal wordt in logica prefix notatie gebruikt



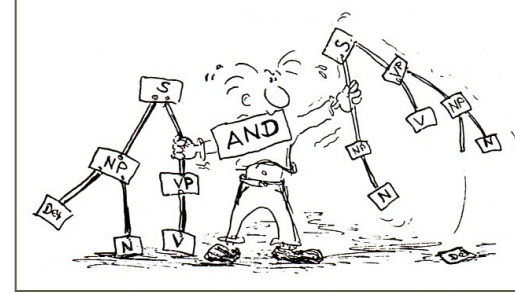
## Definitie

De **formules** in de predikaatlogica zijn als volgt gedefinieerd (inductieve definitie):

- als  $P$  een  $k$ -plaatsige predikaatletter is en  $t_1, \dots, t_k$  zijn termen dan is  $P(t_1, \dots, t_k)$  een formule;
- als  $\varphi$  en  $\psi$  formules zijn, dan zijn ook  $\neg\varphi$ ,  $(\varphi \wedge \psi)$ ,  $(\varphi \vee \psi)$ ,  $(\varphi \rightarrow \psi)$  en  $(\varphi \leftrightarrow \psi)$  formules;
- als  $\varphi$  een formule is en  $x$  een individuele variabele, dan zijn  $\forall x \varphi$  en  $\exists x \varphi$  ook formules;
- niets anders is een formule

Een formule van de vorm  $P(t_1, \dots, t_k)$  heet een **atomaire formule** of **atoom**.

# SYNTAXIS – VOORBEELDEN



## ▶ Voorbeelden atomaire formules

$$P^2(x, a)$$

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

## ▶ Voorbeelden van formules

$$\forall x (A^1(a) \rightarrow \forall y (R^2(x, y) \rightarrow A^1(y)))$$

$$\forall x \exists y \forall z (R^2(z, y) \leftrightarrow \neg Q^2(y, x))$$

## ▶ Zijn geen formules

$$P(\neg x)$$

$$A(x) \forall x$$

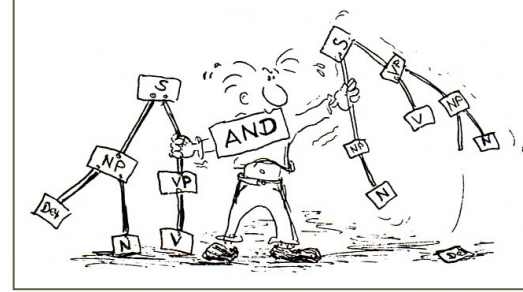
$$\forall x \exists R^2(x, y)$$

We houden in het algemeen de notatie zo eenvoudig mogelijk

- ▶ Zo weinig mogelijk boven indices
- ▶ Zo weinig mogelijk haakjes.



# SYNTAXIS – VOORBEELDEN



## Betekenisvolle voorbeelden

- ▶ Alle honden zijn zoogdieren

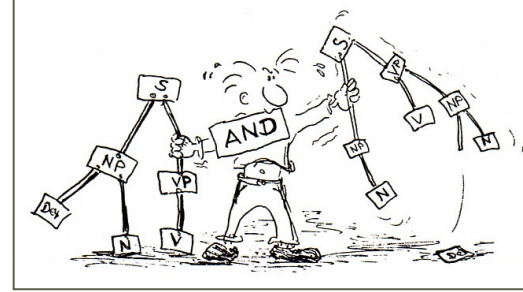
$$\forall x (Hx \rightarrow Zx)$$

- ▶ Sommige voetballers zijn tweevoetig

$$\exists x (Vx \wedge Tx)$$

- ▶ Max ziet iedereen graag

$$\forall x (Mx \rightarrow Gxm)$$



## Betekenisvolle voorbeelden

### ► Niemand kent iedereen

$$\neg \exists x \forall y Kxy$$

### ► Wie iemand kent, kent iedereen

$$\forall x (\exists y Kxy \rightarrow \forall z Kxz)$$

### ► Wie alleen zichzelf kent, wordt door niemand gekend

$$\forall x (\forall y (Kxy \leftrightarrow x=y) \rightarrow \neg \exists z Kzx)$$

## KRACHT VAN DE TAAL

Predikaatlogica is geschikt om wiskundige beweringen te formaliseren, bijv. uit rekenkunde, meetkunde, verzamelingenleer, ...

- ▶ Zie boek voor voorbeelden

Predikaatlogica wordt ook gebruikt als basis voor programmeertalen

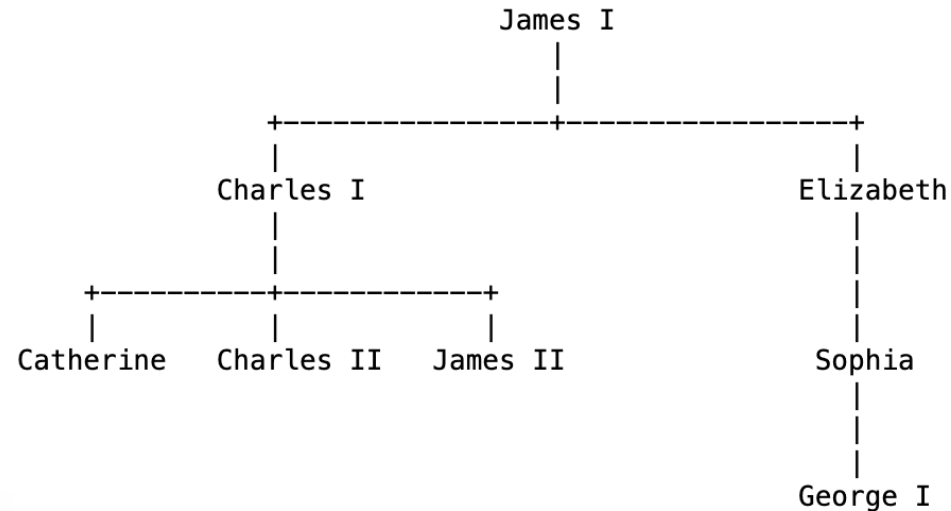
- ▶ De voorbeeldtaal hiervan is PROLOG.

## PROLOG VOORBEELD

```
parent(charles1, james1).
parent(elizabeth, james1).
parent(charles2, charles1).
parent(catherine, charles1).
parent(james2, charles1).
parent(sophia, elizabeth).
parent(george1, sophia).
```

```
male(james1).
male(charles1).
male(charles2).
male(james2).
male(george1).
```

```
female(catherine).
female(elizabeth).
female(sophia).
```



Was George I de ouder van Charles I?

Query: `parent(charles1, george1).`

Wie was Charles I's ouder?

Query: `parent(charles1, X).`

Wie waren de kinderen van Charles I?

Wie was de moeder van Sophia?

Query: `parent(X, charles1).`  
 Query: `parent(sophia, X) and female(X).`

Is Charles II een broer van Catherine?

Query: `parent(charles2, X) and parent(catherine, X) and male(charles).`

Is James een voorouder van George I?

Query: `ancestor(james, georges)`

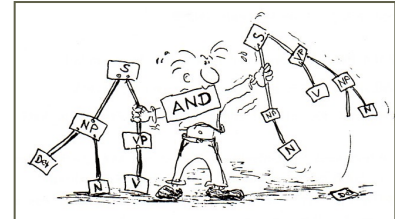
met `ancestor(x, y) :- parent(y, x)`

`ancestor(x, y) :- parent(y, z) and ancestor(x, z)`

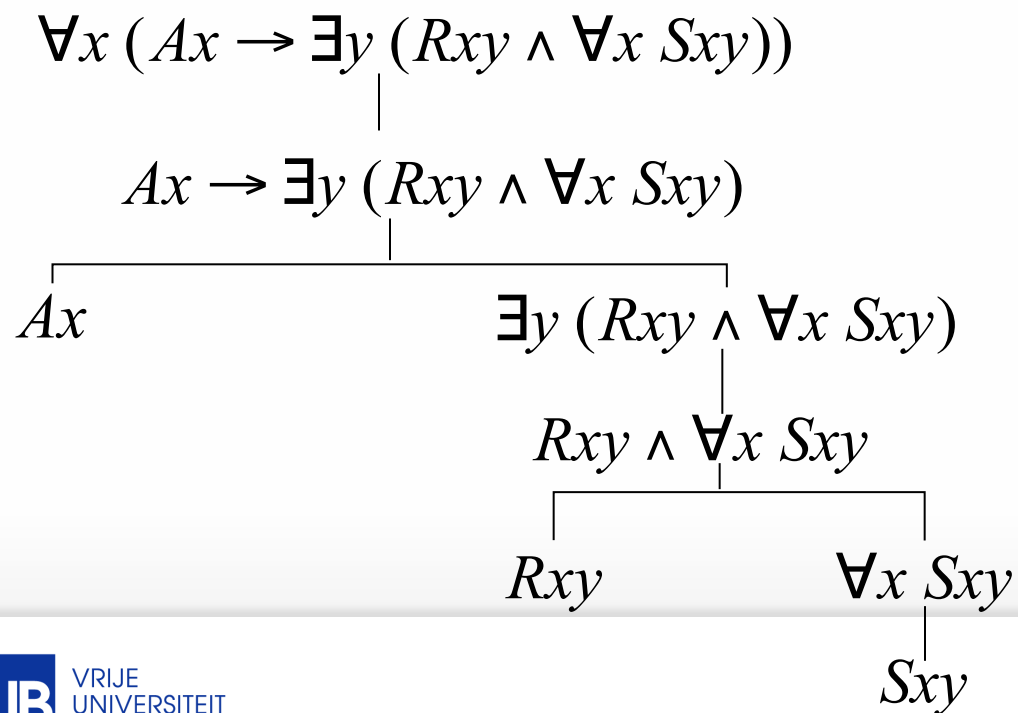


# BEREIK VAN KWANTOREN

## ► Illustratie



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



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

(1)

(2)

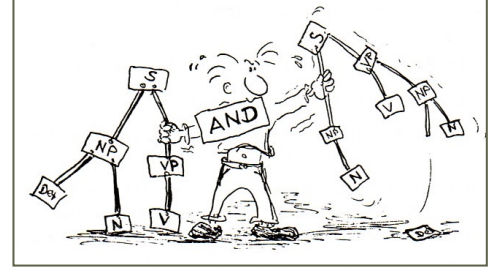
(3)

Bereik van  $\forall x$  (3)

Bereik van  $\exists y$  (2)

Bereik van  $\forall x$  (1)

## VRIJE EN GEBONDEN VARIABELEN



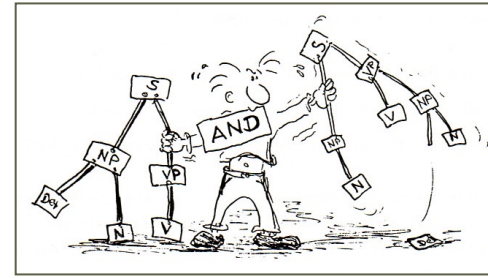
Def. Een *voorkomen* van een variabele  $x$  heet **vrij** als het niet in het bereik van een kwantor  $\forall x$  of  $\exists x$  ligt

Def. Een kwantor  $\forall x$  of  $\exists x$  **bindt** de voorkomens van variabelen  $x$  in zijn bereik, voor zover die nog vrij zijn. Deze voorkomens heten dan **gebonden**

Voorbeeld van gebonden variabelen:

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

## VRIJE EN GEBONDEN VARIABELEN



De vrije variabelen van een formule kunnen beschouwd worden als parameters (zoals in programmeren)

Def. Een **zin** of een **gesloten formule** is een formule zonder vrije variabelen

► Voorbeelden:

$Pa$

$\exists x Px$

$\forall x \exists y (Rxy \vee \exists z (Rxz \wedge Rzy))$

Def. Een formule met vrije variabelen heet een **open formule**

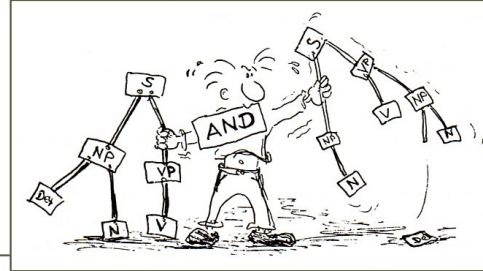
► Voorbeelden

$Rax$

$\exists x Sxy$



# SUBSTITUTIE - DEFINITIE



## Definitie

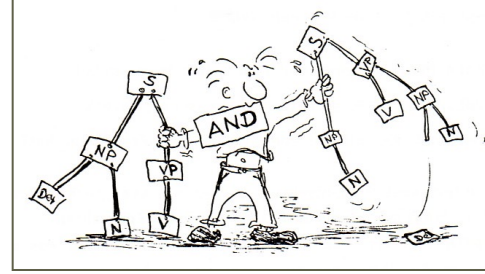
### Substitutie

- ▶ Als  $t, t'$  termen zijn,  $x$  een variabele, dan is  $[t / x]t'$  de term die ontstaat door **elk voorkomen van  $x$**  in  $t'$  te vervangen door  $t$ .
- ▶ Als  $\varphi$  een formule is,  $t$  een term en  $x$  een variabele, dan is  $[t / x]\varphi$  de formule die ontstaat door **elk voorkomen van  $x$  als vrije variabele** in  $\varphi$  te vervangen door  $t$ .  
 $[t / x]\varphi$  wordt ook wel een instantie van  $\varphi$  genoemd

- Voorbeelden

- $[y / x](\forall x (Rx \vee Sx) \vee Pxx) = \forall x (Rx \vee Sx) \vee Pyy$
- $[f(a,b)/z] \exists x (Px \rightarrow Ryz) = \exists x (Px \rightarrow Ry f(a,b))$
- $[y / x](\exists y (y < x)) = \exists y (y < y)$

## "VEILIGE" SUBSTITUTIES



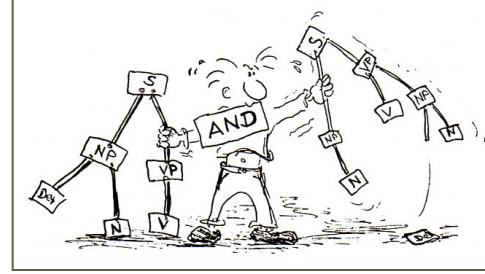
De substitutie van  $t$  voor  $x$  is wel altijd gedefinieerd maar levert soms ongewenste resultaten op (zie vorig voorbeeld)

### *Definitie*

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

- Voorbeelden
  - $f^2zy$  is niet vrij voor  $x$  in  $\exists y Rxy$ , maar wel in  $\exists u Rxu$  of in  $Rxy$

## ALFABETISCHE VARIANT



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

- ▶ Intuïtief gezien verandert hierdoor niets
- ▶  $\exists y (y < x)$  heeft als alfabetische variant:  $\exists z (z < x)$

Wordt gebruikt om substitutie veilig te kunnen uitvoeren:  
Zie vorig voorbeeld:

$f^2zy$  is niet vrij voor  $x$  in  $\exists y Rxy$

Maar wel in  $\exists u Rxu$  (gebonden  $y$  vervangen door  $u$ ).