

# Logisch programmeren 2012

## Opdrachten Week 1

### 1

Maak de oefeningen bij hoofdstukken 1 en 2 van *LPN*. Check je antwoorden met de Oplossingen achter in het boek. Deze opdracht hoef je niet in te leveren.

### 2 Stamboom

Op [deze website](#) vind je de stamboom van een willekeurige familie. Klik op Overzicht voor de complete boom.

**Opgave 1** Vertaal die stamboom, beginnend bij Juliana en Bernhard, in een databank van prolog feiten en regels.

Gebruik feiten voor predicaten `man/1`, `vrouw/1` en `ouder/2`, waarbij je `ouder(X,Y)` interpreteert als `X` is een ouder van `Y`. Je hoeft voor de feiten in je databank de stamboom niet verder uit te vouwen! Klikken op prins Maurits leert dat deze man getrouwd is met prinses Marilène, en dat die weer drie kinderen hebben. Deze leerzame informatie hoef je dus niet bij je feiten op te nemen.

Geef nu regels voor de familierelaties `vader/2`, `moeder/2`, `kind/2`, `zoon/2`, `dochter/2`, `broer/2`, `zuster/2`, `oom/2`, `tante/2`, `neef/2`, `nicht/2`, `grootouder/2`, `grootvader/2`, `grootmoeder/2`, `kleinkind/2`, `kleinzoon/2`, `kleindochter/2`.

Geef ook een definitie voor `gehuwd/2`. Die definitie kan voor een deel uit feiten bestaan; daarnaast wil je ook een regel die uitdrukt dat de relatie `gehuwd/2` omkeerbaar is: als `X` gehuwd is met `Y`, dan is `Y` ook gehuwd met `X`.

Voorzie de definities telkens van een commentaarregel waarin je duidelijk maakt welke interpretatie de predicaten hebben. Bijvoorbeeld (prolog commentaar staat achter het percentteken):

```
% kind/2. kind(X,Y) betekent 'X is een kind van Y'
```

```
kind(X,Y) :- ...
```

Let op: je wil niet dat iemand broer (zus, ...) van zichzelf is. Het ingebouwde predicat `'\='`/2 is handig in dit verband:  $X \neq Y$  betekent dat  $X$  niet unificeerbaar is met  $Y$ . Zorg wel dat je `'\='`/2 op de goede plaats in je regels aanroept.

**Opgave 2** Zet de volgende vragen om in prolog queries.

- Q1 Wie zijn de neven van prins Constantijn?
- Q2 Is prinses Alexia een zus van prinses Mabel?
- Q3 Wie zijn de kleinkinderen van prins Bernhard?
- Q4 Noem de schoondochters van koningin Beatrix.

### 3 $3 \times 3$ woorden

Op de webpagina vind je een lijst van Nederlandse drieletterwoorden, gecodeerd als feiten voor een predicat `drie`/4 zoals hieronder. Het eerste argument is het woord, de volgende drie argumenten de individuele letters waaruit dat woord bestaat.

```
drie(aai,a,a,i).  
drie(aak,a,a,k).  
drie(aal,a,a,l).  
drie(aam,a,a,m).  
drie(aan,a,a,n).
```

**Opgave** Definieer een predicat `drie_bij_drie`/9 waarmee je  $3 \times 3$  vierkanten genereert met in elke rij en in elke kolom een drieletterwoord. Hieronder de bedoelde interpretatie voor de argumenten van `drie_bij_drie`/9.

A1	B1	C1
A2	B2	C2
A3	B3	C3

 $\rightsquigarrow$  `drie_bij_drie(A1,B1,C1,A2,B2,C2,A3,B3,C3)`

### 4 Latijnse vierkanten

Deze opgave is een variatie op de vorige.

Een Latijns vierkant is een  $n \times n$  matrix gevuld met  $n$  verschillende symbolen waarbij elk symbool precies één keer voorkomt in elke rij en in elke kolom. De populaire  $9 \times 9$  sudoku is een speciaal geval van een Latijns vierkant, waarbij de  $9 \times 9$  matrix wordt opgedeeld in  $3 \times 3$  kwadranten; elk kwadrant bevat de 9 gebruikte symbolen.

Hieronder links een gedeeltelijk ingevuld  $3 \times 3$  vierkant. De gebruikte symbolen zijn de getallen 1,2,3. Er is precies één manier om het aan te vullen tot een Latijns vierkant.

1		
	2	
		3

↔

1	3	2
3	2	1
2	1	3

**Opdracht** Schrijf een programma waarmee je voor een symbolset  $\{1, 2, 3\}$ , alle mogelijke  $3 \times 3$  Latijnse vierkanten genereert. Declareer de te gebruiken symbolen als feiten voor een predicaat `symbool/1`. Definieer een predicaat `verschillend/3` voor legitieme combinaties van drie symbolen: combinaties waarbij elk symbool verschillend is. Geef tenslotte een regel voor `latijns_vierkant/9` waarbij je `verschillend/3` gebruikt. De bedoelde interpretatie voor de argumenten van `latijns_vierkant/9` is zoals in de vorige opgave.

A	B	C
D	E	F
G	H	I

↔
`latijns_vierkant(A,B,C,D,E,F,G,H,I)`

## 5 Doordenkertje

Zijn er bij je oplossingen voor `drie_bij_drie/9` ook Latijnse vierkanten? Dat zijn dus oplossingen waarbij de gebruikte woorden zijn opgebouwd uit slechts drie letters, en waarbij elk van die letters precies één keer voorkomt in elke rij en elke kolom.

Je hoeft deze vraag niet in te leveren — hij komt later terug als we over een rijker arsenaal aan programmeerconstructen beschikken.

## 6 Visualisatie

Hieronder nog twee predicaten om je vierkanten als een 3 bij 3 matrix op het scherm af te drukken. (Het ingebouwde `format/2` komt later aan de orde. Met `~w` druk je atomen af, met `~d` getallen.)

```
toon_3letter :-
    drie_bij_drie(A,B,C,D,E,F,G,H,I),
    format("~w~w~w\n~w~w~w\n~w~w~w",
           [A,B,C,D,E,F,G,H,I]).

toon_3letter(A,B,C,D,E,F,G,H,I) :-
    drie_bij_drie(A,B,C,D,E,F,G,H,I),
    format("~w~w~w\n~w~w~w\n~w~w~w",
           [A,B,C,D,E,F,G,H,I]).

toon_lat :-
    latijns_vierkant(A,B,C,D,E,F,G,H,I),
    format("~d~d~d\n~d~d~d\n~d~d~d",
           [A,B,C,D,E,F,G,H,I]).

toon_lat(A,B,C,D,E,F,G,H,I) :-
    latijns_vierkant(A,B,C,D,E,F,G,H,I),
    format("~d~d~d\n~d~d~d\n~d~d~d",
           [A,B,C,D,E,F,G,H,I]).
```

Bijvoorbeeld

```
?- toon_lat(1,_,_,_,2,_,_,_,3).
132
321
213
```

□