

# Tentamen Programmeren in Prolog 6 juni 2001

9.30 – 12.30  
Joris Hulstijn

Dit tentamen bevat vijf opgaven, verdeeld over vier bladzijden. Succes!

|              |         |     |       |      |       |       |
|--------------|---------|-----|-------|------|-------|-------|
| Beoordeling: | I       | II  | III   | IV   | V     | Extra |
|              | 10+10+5 | 5+5 | 7+7+6 | 10+5 | 10+10 | +10   |

## Opgave I: Representatie en redeneren

De volgende tabel geeft een overzicht van de Nederlandse Kampioenschappen Zeilen. Er is precies één wedstrijd voor iedere klasse (type boot of type wedstrijd). Een wedstrijd duurt twee of meer dagen. De organiserende verenigingen zijn aangesloten bij de Koninklijke Nederlandse Watersportbond (KNWV). Wedstrijden met een ‘\*’ tellen mee voor de KNWV beker.

**A.** Ontwerp een geschikte datastructuur voor deze gegevens. Geef de types van de attributen erbij. Welk attribuut is geschikt als *sleutel*? Geef de representatie van de eerste drie wedstrijden (Contender t/m Windsurfen).

| Datum           | Klasse           | Wedstrijdwater      | Organiserende vereniging        |
|-----------------|------------------|---------------------|---------------------------------|
| <b>Juni</b>     |                  |                     |                                 |
| 01-04 *         | Contender        | Noordzee            | Jachtclub Scheveningen          |
| 08-10           | Matchracen       | IJsselmeer          | WSV Hoorn                       |
| 23-24           | Windsurfen       | Schotsman           | Zuid West Zeven                 |
| 23-24           | Kitesurfen       | Noordzee            | Nederlandse Kitesurf Vereniging |
| 29-02           | J/22             | Oosterschelde       | St. Delta Week                  |
| <b>Juli</b>     |                  |                     |                                 |
| 06-08           | Kolibri          | Vinkeveense Plassen | WV Vinkeveen Abcoude            |
| 06-08 *         | Mirror           | Vinkeveense Plassen | WV Vinkeveen Abcoude            |
| 14-16 *         | Schakel          | Sneekmeer           | Kon WV Sneek                    |
| 14-16 *         | FJ               | Sneekmeer           | Kon WV Sneek                    |
| 20-22           | Dart 18          | Katwijk             | WSV Skuytervaart                |
| <b>Augustus</b> |                  |                     |                                 |
| 16-19 *         | Regenboog        | Sneekmeer           | Kon WV Sneek                    |
| 17-20 *         | Pampus           | Braassemermeer      | WV Braassemermeer               |
| 18-21 *         | 16m <sup>2</sup> | Beulakerwijde       | Blauwe Hand Comite '88          |
| . . .           |                  |                     |                                 |

**B.** Geef prolog queries die antwoord geven op de volgende vragen. Geef steeds de definitie van de gebruikte ‘selectorfuncties’.

1. Op welke dagen is het kampioenschap van de Regenboog?
2. Welke wedstrijden tellen er allemaal mee voor de KNWV-beker?
3. Hoeveel wedstrijden zijn er op het Sneekmeer?
4. Welke wedstrijden vallen er (deels) in de maand juli?

**C.** Het publiek bezoekt graag spectaculaire wedstrijden, of wedstrijden waar het 's avonds gezellig is. Daarom willen we redeneren over wedstrijden.

Hoe zien de volgende redeneerregels er uit in Prolog syntax?

1. Na afloop van wedstrijden op het Sneekmeer is het gezellig.
2. Windsurfwedstrijden of wedstrijden van de Dart 18 zijn spectaculair.

## Opgave II: Cut

De ‘Cut’ wordt vaak gebruikt voor het weergeven van exclusieve regels, met een algemeen geldende ‘default’ regel aan het eind. Gegeven is een functie voor het bepalen van uitkeringen aan de hand van feiten over personen.

```
% uitkering(+Persoon, -UitkeringsType)
uitkering(X, invaliditeits_uitkering) :-      % regel 1
    invalide(X), !.
uitkering(X, pensioen) :-                      % regel 2
    boven_65(X),
    premie_betaald(X), !.
uitkering(_, bijstand).                       % ‘default’
```

**A.** Zijn de cuts in regel 1 en 2 gebruikt als rode of groene cuts? Motiveer je antwoord.

**B.** Kan je dit stuk programma herschrijven zonder een cut? Zo ja, hoe dan? Zo nee, waarom niet?

## Opgave III: Lijsten

**A.** Definieer het predikaat `splits(+L, -Even, -Oneven)` dat een lijst van getallen splitst in twee lijsten met even en oneven getallen. Het predikaat `even/1` is gegeven, namelijk `even(X) :- X mod 2 == 0`.

```
?- splits( [1,2,3,4,5,6], E, O ).
E = [2,4,6],
O = [1,3,5];
No.
```

**B.** Stel, we representeren verzamelingen getallen door middel van gesorteerde lijsten (GL). Gesorteerde lijsten mogen geen dubbele elementen bevatten.

Definieer het predikaat `insert( +Elt, +GL, -GL1)` dat een element op de juiste plek in een gesorteerde lijst plaatst.

```
?- insert( 4, [1,3,5,7], [1,3,4,5,7] ).
Yes.
?- insert( 3, [1,3,5,7], [1,3,5,7] ). % 3 zat er al in
Yes.
```

**C.** Definieer het predikaat `gemiddelde(+L, -Gem)` dat het gemiddelde uitrekent van een lijst van getallen.

```
?- gemiddelde([1,2,3,4],Gem).
Gem = 2.5 ;
No
```

## Opgave IV: Zoeken

Gegeven is de volgende zoekruimte.

```
trans(amsterdam_cs,    amersfoort,    30). % treinverbinding
trans(amsterdam_amstel, amsterdam_cs, 10). % met afstand in km
trans(amsterdam_amstel, utrecht,      30). % (een richting)
trans(utrecht,         amersfoort,    20).
trans(amersfoort,      groningen,    140).
trans(amersfoort,      enschede,      120).

verbinding(A, B, Afstand) :-                % treinverbinding
    trans(A, B, Afstand) ;                  % (beide richtingen)
    trans(B, A, Afstand).

knoop(A) :-                                  % toestanden
    verbinding(A,_,_).

begin(enschede).                             % begintoestand
eind(amsterdam_amstel).                      % eindtoestanden
eind(amsterdam_cs).
```

**A.** Definieer een predikaat dat met behulp van het *breadth first* algoritme een pad bepaalt van de begintoestand naar een van de eindtoestanden, en daarbij de totale lengte van het pad uitrekent.

Geef aan hoe je predikaat moet worden aangeroepen.

**B.** Wanneer is een zoekalgoritme toegestaan (admissible)? (max 4 regels)

Is het breadth first algoritme toegestaan? Motiveer je antwoord (max 4 regels).

Op de volgende bladzijde staat vraag V.

## Opgave V: Grammatica's

Gegeven is de volgende grammatica.

```
% Grammatica
zin  --> znw_g, ww_g.      % onderwerp en gezegde
znw_g --> naam.
znw_g --> lidw, znw.
ww_g  --> intrans_ww.      % zonder lijdend voorwerp
ww_g  --> trans_ww, znw_g. % met lijdend voorwerp

naam --> [jan];[marie].
lidw --> [de];[het];[een].
znw  --> [boek];[huis];[cactus],[meneer].
intrans_ww --> [loopt];[fietst];[zingt].
trans_ww  --> [koopt];[haat].
```

**A.** Bepaal of de volgende prolog aanroepen zullen slagen, of niet. Teken de ontledingboom (parse tree) van de geaccepteerde expressies erbij.

1. `zin( [de, boek, koopt, marie], []).`
2. `phrase(zin, [koopt, jan, een, boek]).`
3. `ww_g( [fietst, een, boek], []).`
4. `zin( [de, meneer, koopt, een, boek], []).`
5. `zin( [de, meneer, fietst, een, boek], [een,boek]).`

**B.** De bovenstaande grammatica accepteert en genereert ongewenste zinnen. Pas de grammatica zo aan, dat de volgende expressies niet meer voorkomen.

1. "het meneer", "de boek"
2. "het boek zingt", "het boek koopt een huis", "jan koopt marie"