

Tentamen Prolog

31 mei 2007

De eindbeoordeling is "het aantal punten + 10" gedeeld door 10.

Vraag:	1	2a	2b	2c	3a	3b	3c	4a	4b	5
Punten:	10	10	5	5	10	5	5	10	10	20

Veel succes!

Vraag 1: termen en lijsten

Geef voor elk van de volgende goals aan of ze slagen of falen. Geef bij slagende goals aan wat de resulterende variabele-binding(en) is, geef bij falende goals een korte uitleg.

1. $f(X, g(A, B), Z) = f(A, g(1, B), X)$.
2. $f(X, g(A, B), Z) = f(A, g(h(X), B), 1)$.
3. $f(A, 2, 3) = \dots [f, 1, 2, X]$.
4. $[1, 2, [3, 4]] = [X|Y]$.
5. $[1, 2, [3, 4]] = [X, Y|Z]$.
6. $[1, 2|3] = [X, Y]$.
7. X is $5+7$.
8. $12 = 5+7$.
9. $f(X, Y) == f(X, X)$.
10. $Y = 5+7, X$ is $Y, f(X, Y) = f(X, X)$.

Vraag 2: Programma executie

Beschouw het onderstaande programma, en het doel $?:- f(1,Q)$.

```
f(X,Y) :- g(X), h(X,Z), i(Z,Y).  
g(0).  
g(1).  
g(2).  
g(3).  
h(A,B) :- B is 5*A.  
h(A,B) :- B is 10*A.  
i(A,B) :- B is A/10, integer(B).  
i(A,B) :- B is A/5.
```

- (a) Beschrijf nauwkeurig de executie van het programma tot aan het eerste antwoord van Prolog. Benoem daarbij voor elke executie stap welke variabelen gebonden worden, welke clause en welk doel in die clause uitgevoerd worden, wat het resultaat van het clause of doel is, en hoe de berekening vervolgd wordt na de stap.
- (b) Na afloop van de zojuist door jou beschreven berekening typt de gebruiker een ';'. Vervolg je beschrijving tot aan het volgende antwoord van Prolog.
- (c) Vervolgens wordt het programma gemodificeerd door een '!' aan te brengen vlak voor de aanroep van $i(Z,Y)$ in de clause voor $f(X,Y)$. Wat voor verschil zou dat maken voor je eerder gegeven antwoorden?

Vraag 3: cuts en negatie

Cut en negatie zijn sterk aan elkaar verwant.

- (a) Beschrijf hoe je cut kunt gebruiken om in Prolog negatie te definiëren.

- (b) Beschrijf hoe je negatie kunt gebruiken om een cut uit een programma te verwijderen. Illustreer je antwoord aan de hand van de code voor het predicaat `max(X,Y,Max)`, dat het maximum van twee getallen `X` en `Y` uitrekent.
- (c) Beschrijf het verschil tussen red cuts en green cuts. Illustreer weer de verschillen aan de hand van `max(X,Y,Max)`.

Opgave 4: Ontwerp van data-structuren

Van een patient willen we de volgende gegevens opslaan:

- de naam, als string
- het adres, bestaande uit:
 - straat (een string),
 - huisnummer (een getal),
 - postcode (een atoom) en
 - plaats (een string)
- geboortedatum, bestaande uit:
 - dag,
 - maand en
 - jaar (elk een getal)
- de ziektegeschiedenis van de patient, bestaande uit een lijst dokterbezoeken met bij elk dokterbezoek
 - een ziekte (een string) en
 - de datum waarop de patient voor die ziekte bij de arts geweest is. Datum is weer als boven.

(a) Bedenk een geschikte datastructuur om zulke gegevens in Prolog op te slaan. Geef aan hoe de volgende patient in jouw datastructuur zou worden weergegeven:

- Martin Hardman,
- wonende op de Wittenburgergracht 23, 1018MT, Amsterdam,
- geboren op 29 Oktober 1960,
- bezocht op 7 April 1994 de dokter voor een gebroken been, en daarvoor op 12 November 1985 voor migraine.

(b) Schrijf Prolog queries die gegeven een database van patienten in het door jou bedachte formaat de volgende vragen beantwoorden:

- Is er een patient die 1960 als geboortejaar heeft?
- Is er een patient die is geboren in de periode 1940-1945?
- Heeft Peter Hardman in 1990 de dokter bezocht?
- Hoeveel patienten staan er in de database?
- Wat zijn de namen van alle patienten die in het huidige jaar meer dan 2 bezoeken aan de dokter hebben gebracht?

Opgave 5. Recursie en term-manipulatie

Schrijf een predicaat `match(Term1,Term2)`, die een liberale vorm van unificatie implementeert. Het predicaat `match` controleert of de structuur van twee termen hetzelfde is (dat wil zeggen: overal dezelfde functoren met dezelfde aantallen argumenten), maar negeert eventuele verschillen tussen de atomen in beide termen.

Voorbeelden van het gewenste gedrag van het predicaat `match` zijn:

- `match(f(a),f(b))` slaagt
- `match(f(a,g(b)),f(c,g(e)))` slaagt
- `match(f(a,g(b)),f(c,h(e)))` faalt (want `g` en `h` zijn verschillende functoren)
- `match(f(X,g(b,c)),f(d,g(e,Y)))` slaagt met $X = d$ en $Y = c$.

Einde tentamen