

Hertentamen Logisch Programmeren (LIX003B05, 29 maart 2018)

Opgave 1. **Unification** (17 points (*punten*))

How does Prolog answer the following queries (*Welke antwoorden geeft Prolog op de volgende queries*)?

- (a) `?- [een,X,drie,[]] = [een,twee,drie].`
- (b) `?- [een,X,drie|[]] = [een,twee,drie].`
- (c) `?- sort([1,2,3,4]) = sort([4,2,3,1]).`
- (d) `?- [een,twee|[drie,vier]] = [een,twee,drie,vier].`
- (e) `?- [Hey] = [hey,hey].`
- (f) `?- [een,twee|Drie] = [een,twee,drie].`
- (g) `?- [Hey,Hey] = [hey(A),hey(b)].`
- (h) `?- [een,twee|Drie] = [een,twee,3,4].`
- (i) `?- [hey,hey] = HeyHey.`
- (j) `?- [A|[]] = [hey].`

Opgave 2. **Split** (12 points (*punten*))

The following Prolog database is given (*gegeven is de volgende Prolog database*):

```
split([X|L3],L4,L5):- X = 0, !, split(L3,L4,L5).
split([X|L0],[X|L1],L2):- X > 0, !, split(L0,L1,L2).
split([X|L6],L7,[X|L8]):- X < 0, !, split(L6,L7,L8).
split([],[],[]).
```

- (a) How many clauses are there in this database (*hoeveel clauses zijn er*)?
- (b) How will Prolog respond to the query (*hoe reageert Prolog op de volgende query*):
`?- split([0,4,-5,9,0,4,9],X,Y).`
- (c) There are four cuts in this database (*er bevinden zich vier snedes in deze database*). For each cut, describe what effect it has (*beschrijf het effect van elke snede*).

Opgave 3. **Harmonic Mean** (12 points (*punten*))

The harmonic mean h of two numbers a and b is computed as: $h = (2 * a * b) / (a + b)$ (*het harmonisch gemiddelde h van twee getallen a en b wordt als volgt berekend: $h = (2 * a * b) / (a + b)$*).

- (a) Write a predicate `harmean1/3` that computes the harmonic mean of two numbers (*schrijf een predikaat `harmean1/3` dat het harmonisch gemiddelde van twee getallen berekend*). For instance (*bijvoorbeeld*):
`?- harmean1(0.6, 0.4, X).`
`X=0.48`
`yes`
- (b) Write a predicate `harmean2/3` that computes the harmonic mean of two numbers between 0 and 1 (*schrijf een predikaat `harmean2/3` dat het harmonisch gemiddelde van twee getallen tussen 0 en 1 berekend*). For instance (*bijvoorbeeld*):
`?- harmean2(3, 5, H).`
`no`
`?- harmean2(0.3, 0.4, H).`
`H = 0.342857142`
`yes`

Opgave 4. **Reverse Order** (8 points (*punten*))

Write a recursive predicate `revorder/1` that is true if and only if its argument is a (non-empty) ordered list of integers in decreasing order (*Schrijf een recursief predikaat `revorder/1` dat alleen waar is als zijn argument een (niet-lege) lijst van gehele getallen is die in aflopende grootte gesorteerd is*). Example queries (*voorbeeldqueries*):

```
?- revorder([1,4,8,12]).  
no
```

```
?- revorder([7,5,2]).  
yes
```

```
?- revorder([]).  
no
```

```
?- revorder([25]).  
yes
```

```
?- revorder([4,2,2,1]).  
no
```

Opgave 5. **Occurs Check** (5 points (*punten*))

What is meant by *occurs check*? Wat wordt er met de *occurs check* bedoeld?

Opgave 6. **Balanced Words** (10 points (*punten*))

A balanced word is a word that has an equal number of vowels and consonants. An example is “boek”, because it contains two vowels (“o” and “e”) and two consonants (“b” and “k”). *Een gebalanceerd woord is een woord dat uit evenveel klinkers en medeklinkers is samengesteld. Een voorbeeld is “boek”, want dit woord heeft twee klinkers (“o” and “e”) en twee medeklinkers (“b” and “k”).*

Write a predicate `balanced/1` that is true if its argument is a balanced word. Words are represented by lists of characters. (*Schrijf een predikaat `balanced/1` dat waar is als zijn argument een gebalanceerd woord is. Woorden worden in lijsten van karakters weergegeven.*)

Opgave 7. **Allen’s Interval Algebra** (10 points (*punten*))

Given is the following Prolog database with definition of some of Allen’s interval algebra (*gegeven is de volgende Prolog database met definities van enkele relaties van Allens intervalalgebra*):

```
meets(interval(_,B),interval(B,_)).  
contains(interval(A,B),interval(C,D)):- A < B, C < D, A > C, B > D.  
during(A,B):- contains(B,A).  
interval(interval(A,B)):- A < B.  
equal(A,A):- interval(A).
```

- (a) How many clauses does this database contain (*hoeveel clauses*)?
- (b) How many rules are there in this database (*hoeveel regels*)?
- (c) How many facts are there in this database (*hoeveel feiten*)?
- (d) Which predicates are there defined in this database (*Welke predikaten zijn er in deze database gedefinieerd*)?
- (e) How many recursive predicates are defined in this database (*Hoeveel recursieve predikaten zijn er in deze database gedefinieerd*)?
- (f) How many tail-recursive predicates are defined in this database (*Hoeveel staart-recursieve predikaten zijn er in deze database gedefinieerd*)?

Opgave 8. **Memory** (4 points (*punten*))

What is meant by memoisation in Prolog, and what Prolog predicates are used for this? (*Wat wordt er in Prolog met memoisation bedoeld, en welke ingebouwde Prolog predikaten zijn hiervoor?*)

Opgave 9. **Counting** (6 points (*punten*))

How many elements does each of the following Prolog lists contain? (*Hoeveel elementen bevat elk van de volgende Prologlijsten?*)

- (a) `[een,twee,drie|[]]`
- (b) `[een,twee,drie,[],[]]`
- (c) `[een|[twee,drie]]`
- (d) `[een,twee,X]`
- (e) `[een,twee|X]`

Opgave 10. **Search Trees** (16 points (*punten*))

Given is the following database in Prolog: (*Gegeven is de volgende Prolog database*):

```
rebmem([_|L],X):- rebmem(L,X).  
rebmem([X|_],X).
```

Draw the complete search trees for the following queries: (*Teken de volledige zoekbomen voor de volgende queries*):

- (a) `?- rebmem([john,paul,george],B).`
- (b) `?- rebmem([john,paul],ringo).`
- (c) `?- rebmem([],B).`