Tentamen Logisch Programmeren (LIX003B05, 24 januari 2019)

Opgave 1. **The IT Crowd** (4 points (*punten*))

Specify whether the following Prolog terms are atoms or variables. (*Geef van de volgende vijf Prologtermen aan of het atomen of variabelen zijn.*)

(a) `jen`

(b) `Moss`

(c) `_Chris`

(d) `'CLICK'`

Opgave 2. **If all else fails...** (14 points (*punten*))

Which of the following unification attempts fail, and which succeed? (*Welke van de volgende unificatie-pogingen slagen, en welke falen*)?

(a) `?- a = A.`
(b) `?- b = b(0).`
(c) `?- A = aa.`
(d) `?- b(A) = a(B).`
(e) `?- [a|[]] = [a].`
(f) `?- [b,[]] = [b].`
(g) `?- [a|[a]] = [a,a].`
(h) `?- [b,[b]] = [b,b].`
(i) `?- [X|[a]] = [a,X].`
(j) `?- [b,[B]] = [b,b].`
(k) `?- [a|[a|[a]]] = [a|[a,a]].`
(l) `?- [[b]] = [b].`
(m) `?- [a,b,c] = reverse([c,b,a]).`
(n) `?- a(b(C)) = a(B).`

Opgave 3. **Something to remember...** (15 points (*punten*))

Write a predicate `remember/2` that is true if and only if its first argument (any Prolog term) appears at least twice in its second argument (a Prolog list). (*Schrijf een predikaat `remember/2` dat waar is als zijn eerste argument (een willekeurige Prolog-term) tenminste twee keer voorkomt in zijn tweede argument (een Prolog-lijst)*):

```
?- remember(c,[a,b,c,d,e,c,f]).
yes

?- remember(f,[a,b,c,d,e,c,f]).
no

?- remember(e,[a,b,c,d,e,c,f,e,e,e]).
yes
```

Opgave 4. **Measure twice and cut once...** (10 points (*punten*))

Write a predicate `twice/2` that is true if and only if its first argument (any Prolog term) appears exactly twice in its second argument (a Prolog list). (*Schrijf een predikaat `twice/2` dat waar is als zijn eerste argument (een willekeurige Prolog-term) precies twee keer voorkomt in zijn tweede argument (een Prolog-lijst)*):

```
?- twice(c,[a,b,c,d,e,c,f]).
yes

?- twice(f,[a,b,c,d,e,c,f]).
no

?- twice(e,[a,b,c,d,e,c,f,e,e,e]).
no
```

Opgave 5. **The winner takes it all...** (12 points (*punten*))

Assume we have a database of results of beerpong games played by members of ASCI Groningen. The pairings were not arranged in any systematic way, so each player just played some other players. The results are represented as follows in this Prolog programme:

```
beat(pauline,nick).
beat(leon,nick).
beat(kamil,pauline).
```

Define a predicate `class/2` that assigns categories to players. We have just three categories:

winner : a player who won all his or her games;

fighter : a player who won some and lost some games;

loser : a player who lost all games.

For instance:

```
?- class(pauline,X).
X = fighter.
true

?- class(X,winner).
X = leon.
true
```

Opgave 6. **Fibonacci** (11 points (*punten*))

The Fibonacci sequence is: 1, 1, 2, 3, 5, 8, 13, ... Each number in the sequence (except for the first two), is the sum of the previous two numbers. Consider the following Prolog program which computes the Nth Fibonacci number F:

```
fib(N,F):-
   N > 2,
   N1 is N - 1,
   fib(N1,F1),
   N2 is N - 2,
   fib(N2,F2),
   F is F1 + F2.
fib(2,1).
fib(1,1).
```

Now answer the following questions:

(a) How does Prolog answer the query `?- fib(4,X).`

(b) How does Prolog answer the query `?- fib(0,X).`

(c) How many clauses does this program consists of?

(d) How many facts does this program consists of?

(e) How many rules does this program consists of?

(f) How many predicates are defined in this program?

(g) How many recursive predicates are defined in this program?

(h) How many tail-recursive predicates are defined in this program?

(i) What is the best place in this programme to add a cut?

(j) Is this is a green or a red cut?

(k) Which built-in Prolog predicate would you use to add memoisation to this program?

Opgave 7. **A shift in the wind...** (10 points (*punten*))

Write a predicate `shift/2` with two lists as arguments, so that the second list is shifted rotationally by one element to the left. For instance:

```
?- shift([1,2,3,4,5],L).
L = [2,3,4,5,1].
true
```

If you want to use built-in predicates like `member/2` or `append/3` you will need to give the definition. If you don't use such predicates then you get a bonus of five points.

Opgave 8. **Don't bark up the wrong tree...** (15 points (*punten*))

Consider the following Prolog predicate:

```
bark(X,Y):-
   X = [_|L],
   wrong(L,Y).

wrong([X|L1],[X|L2]):-
   wrong(L1,L2).

wrong([_],[]).
```

(a) Draw the search tree for `?- bark([a],A).`

(b) Draw the search tree for `?- bark([a,b],A).`

(c) Draw the search tree for `?- bark([a,b,c],A).`

Opgave 9. **Occurs Check** (5 points (*punten*))

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

Opgave 10. **Elfstedentocht** (4 points (*punten*))

How many elements are there in the following five lists?

(a) `[workum|[bolsward,harlingen,franeker,dokkum,leeuwarden]]`

(b) `[[]|[]]`

(c) `[1909,1912,1917,X]`

(d) `[1909,1912,1917|X]`

This exam has 10 questions. Total number of points: 100.
Dit tentamen bevat 10 opgaven. Puntentotaal: 100.

Docent: Johan Bos

Tweede lezer: Lasha Abzianidze