

Tentamen Logisch Programmeren (LIX003B05, 1 februari 2023)

Opgave 1. Elfstedentocht (7 points)

How many elements are there in the following six lists?

Hoeveel elementen bevatten de volgende zes Prologlijsten?

- (a) []
- (b) [[]|[]]
- (c) [leeuwarden,sneek,ijlst,sloten,stavoren|[]]
- (d) [workum|[bolsward,harlingen,dokkum,leeuwarden]]
- (e) [1909,1912,1917|X]
- (f) [1909,1912,1917,X]

Opgave 2. Unification (12 points)

Which of the following queries give a positive answer in Prolog, and which yield a negative result? Give the variable bindings for each positive answer, and explain why there is no unification for the negative answers. *Op welke van de volgende queries geeft Prolog een positief antwoord, en op welke een negatief? Geef bij positieve antwoorden ook de waarde van eventuele variabelen aan. Leg bij negatieve antwoorden uit waarom geen unificatie plaats kan vinden.*

- (a) `?- eye = Tooth.`
- (b) `?- e(Ey) = t(Ooth).`
- (c) `?- x(Eye) = x(Tooth).`
- (d) `?- eyes([1+1]) = eyes([2]).`
- (e) `?- [eye,tooth] = sort([tooth,eye]).`
- (f) `?- sort([eye,tooth]) = sort([Tooth,Eye]).`

Opgave 3. An odd split (15 points)

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

```
split([], [], []).
split([X|L0], [X|L1], L2):- X > 0, !, split(L0, L1, L2).
split([X|L0], L1, L2):- X = 0, !, split(L0, L2, L1).
split([X|L0], L1, [X|L2]):- X < 0, !, split(L0, L1, L2).
```

- (a) Is `split/3` a recursive predicate? Is it tail-recursive?
Is split/3 een recursief predicaat? Is het staart-recursief?
- (b) How will Prolog respond to the query (*hoe reageert Prolog op de volgende query*):
`?- split([4,-5,9,6,-2,4],X,Y).`
- (c) How will Prolog respond to the query (*hoe reageert Prolog op de volgende query*):
`?- split([4,-5,0,-2,4],X,Y).`
- (d) There are three cuts in this database (*er bevinden zich drie snedes in deze database*). Classify the behaviour of each cut as red, green, or blue (*classificeer het gedrag van elke snede als rood, groen of blauw*).
- (e) Change the definition of `split/3` in such a way that it renders a green cut into a red cut (*verander de definitie van split/3 zodanig dat een groene snede rood wordt*).

Opgave 4. Mean (10 points)

Write a Prolog program that computes the arithmetic mean of a group of numbers, i.e. the sum of all values divided by the number of values (*schrijf een Prolog programma dat het gemiddelde van een groep getallen berekent, dwz. de som van alle waarden gedeeld door het aantal waarden*). Your program should include a wrapper predicate `mean/2`. All other predicates that you want to use need to be defined. You get a bonus if you restrict the number of predicates (including the wrapper) to two.

```
?- mean([3,7,2],Mean).
```

```
Mean = 4.
```

```
true
```

```
?- mean([3,7,2],5).
```

```
false
```

```
?- mean([3,6,1,2],M).
```

```
M = 3.
```

```
true
```

Opgave 5. Marsupial (10 points)

A marsupial word is a word that carries another word within it without transposing its letters. For instance, the word "scary" carries the word "car", and the word "interesting" carries the words "interest", "rest", and "sting". In the Prolog database below words are represented by lists of characters.

Een buideldierwoord is een woord dat een ander woord met de letters in dezelfde volgorde in zich draagt. Bijvoorbeeld, het woord "chips" draagt het woord "hip", en het woord "onmiddellijk" draagt de woorden "middel", "del" en "lijk". Woorden zijn in deze Prolog database op de volgende manier weergegeven.

```
word([c,a,r]).
```

```
word([r,e,s,t]).
```

```
word([s,c,a,r,y]).
```

```
word([s,t,i,n,g]).
```

```
word([i,n,t,e,r,e,s,t,i,n,g]).
```

```
word([i,n,t,e,r,e,s,t,i,n,g,l,y]).
```

Write a predicate `marsupial/1` that is true if its argument is a marsupial word, given the words in the database above. See some example queries below. If you want to make use of list processing predicates such as `member/2` or `append/3` then give their definition as well. *Schrijf een predikaat marsupial/1 dat waar is als zijn argument een buideldierwoord is op basis van de woorden in deze database. Hieronder staan wat voorbeelden van queries. Als je gebruik wilt maken van predikaten zoals member/2 of append/3 geef dan ook de definitie hiervan.*

```
?- marsupial([r,e,s,t]).
```

```
false
```

```
?- marsupial([i,n,t,e,r,e,s,t,i,n,g]).
```

```
true
```

```
?- marsupial([s,t,r,e,s,s]).
```

```
false
```

```
?- marsupial([i,n,t,e,r,e,s,t,i,n,g,l,y]).
```

```
true
```

Opgave 6. **Pick** (25 points)

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

```
pick(X,X).
pick(X,[X|L],L):-!.
pick(K,[X|L],[Y|M]):-pick(X,Y),pick(K,L,M).
```

- How many clauses does this database contain (*hoeveel clausules*)?
- How many rules are there in this database (*hoeveel regels*)?
- How many facts are there in this database (*hoeveel feiten*)?
- Which predicates are defined in this database (*Welke predikaten zijn in deze database gedefinieerd*)?
- Draw the complete search trees for the following three queries (*teken de volledige zoekbomen voor de volgende drie queries*):

- i. ?- pick(A, [a,b,c], W) .
- ii. ?- pick(b, [a,b,c], W) .
- iii. ?- pick(b, W, [a,c]) .

Opgave 7. Cryptarithmic (10 points)

In the following mathematical equation each letter represents a digit (0, 1, 2, 3, 4, 5, 6, 7, 8, or 9), and each letter is a distinct digit. Write a Prolog programme that identifies the value of each letter such that the sum is satisfied. *In de volgende wiskundige vergelijking vertegenwoordigt elke letter een cijfer (0, 1, 2, 3, 4, 5, 6, 7, 8 of 9), en elke letter is een ander cijfer. Schrijf een Prolog-programma dat de waarde van elke letter identificeert zodanig dat de som klopt.*

$$\begin{array}{r} \text{T} \text{O} \\ + \text{G} \text{O} \\ \hline = \text{O} \text{U} \text{T} \end{array}$$

Opgave 8. Forever Young (11 points)

Consider the following database with the predicate `dob/2` which first argument is a name of a person and its second argument their date of birth (*gegeven is de volgende database met het predikaat `dob/2` waarbij het eerste argument een naam van een persoon is en het tweede argument zijn of haar geboortedatum*):

```
dob('Mick', 1943).
dob('Charlie', 1941).
dob('Ronnie', 1947).
dob('Paul', 1942).
```

- Define a predicate `eldest/1` that is true if and only if its argument unifies with the eldest person in the database, making use of Prolog negation `\+`.
- Define a predicate `eldest/1` that is true if and only if its argument unifies with the eldest person in the database, making use of the Prolog `setof/3` predicate.

This exam has 8 questions. Total number of points: 100.
Dit tentamen bevat 8 opgaven. Puntentotaal: 100.