

Herkansing Programmeren in Prolog: augustus 2005

VRAAG	AANTAL PUNTEN
1.Unificaties	15
2.Kennisrepresentatie	20
3.Programma	15
4.Cut	20
5.Definite clause grammars	20
<i>Cijfer :</i>	$(aantal + 10)/10$

Veel succes!

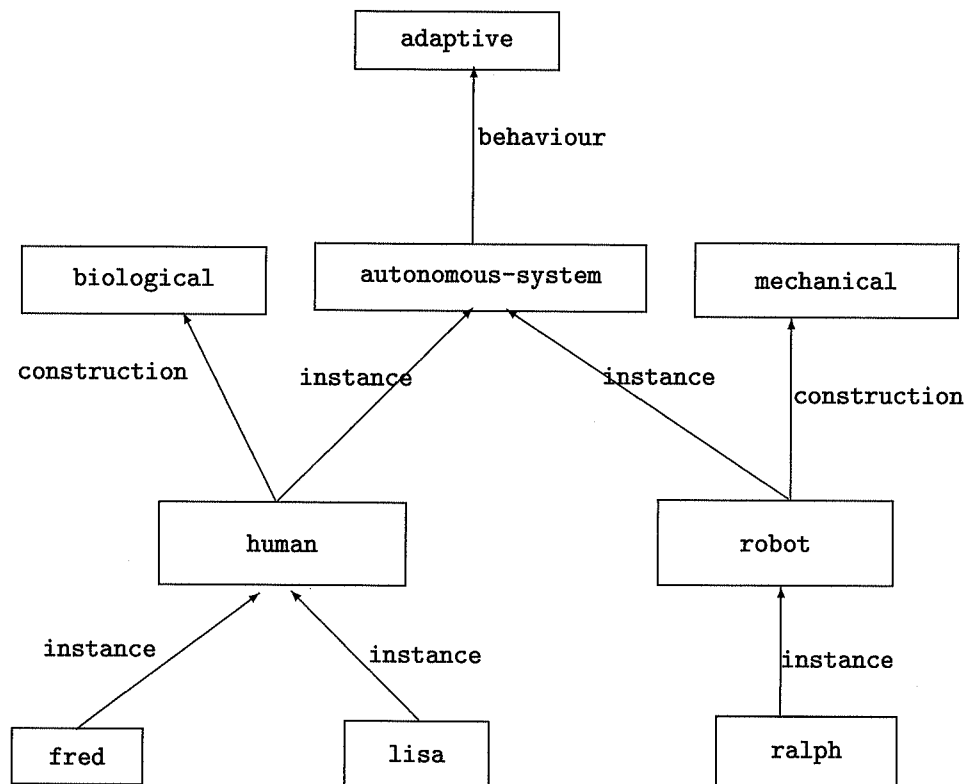
Vraag 1: Unificaties (15 punten)

Wat is het antwoord van de Prolog interpreter op de onderstaande queries?

1. ?- [X,Y,Z] = [john,likes,fish].
2. ?- [cat] = [X|Y].
3. ?- [X,Y|Z]=[mary,likes,wine].
4. ?- [[the,Y|Z] = [[X,hare],[is,here]].
5. ?- [X,Y|Z,W] = [1,2;3,4].
6. ?- [golden|T]=[golden,norfolk].
7. ?- [vale,horse]=[horse,X].
8. ?- [white|Q]=[P|horse].
9. ?- a(b,C,d(e,F,g(h,i,J)))=a(F,c,d(J,F,g(h,i,J))).
10. ?- f(1,2)=X(Y,Z).
11. ?- Y = 3, X is 3 + Y.
12. ?- X is 3 + Y, Y = 3.
13. ?- X is 12, X := 6 + 6.
14. ?- X = 12, X = 6 + 6.
15. ?- f(A,B) == f(X,Y).

Vraag 2: Kennisrepresentatie (Semantische netwerken) (20 punten)

Gegeven het onderstaande semantische netwerk.



Vraag 2a: (5 punten)

Geef een representatie in Prolog voor het bovenstaande semantische netwerk.

Aan de feiten uit vraag 2a, wordt de volgende procedure toegevoegd (jullie bekend uit het boek):

```
% eerste clause van fact
fact(Fact):-
    Fact,! .

%tweede clause van fact
fact(Fact):-
    Fact =.. [Rel,Arg1,Arg2],
    instance(Arg1,SuperArg),
    SuperFact =.. [Rel,SuperArg,Arg2],
    fact(SuperFact).
```

Vraag 2b: (5 punten)

Geef een voorbeeld van een query die slaagt door de eerste clause van “fact”. Gebruik voor het voorbeeld het gegeven semantische netwerk.

Vraag 2c: (10 punten)

Geef een voorbeeld van een query die slaagt door de tweede clause van “fact”. Geef ook de afleidingsboom van deze query, en geef daarbij ook aan indien er takken worden weggesneden ivm. de cut operator.

Gebruik weer voor het voorbeeld het gegeven semantische netwerk.

Vraag 3: Cuts (20 punten)

Vraag 3a:

Schrijf de equivalente logische expressies voor de volgende logische programma's:

```
% programma 1.  
p :- a,b.  
p :- c.
```

```
%programma 2  
p :- a,! ,b.  
p :- c.
```

```
% programma 3  
p :- c.  
p :- a,! ,b.
```

Vraag 3b:

Leid uit opgave (a) af of de cuts gebruikt in programma 2 en programma 3 groen of rood zijn ten opzichte van programma 1. Motiveer je antwoord.

Vraag 3c:

Gegeven het onderstaande Prolog programma met de volgende feiten:

```
p(a).  
p(b).  
q(a,1).  
q(a,2).  
q(b,3).  
q(b,4).  
r(1,1).  
r(1,2).  
r(2,3).  
r(2,4).  
r(3,5).  
r(3,6).  
r(4,7).  
r(4,8).
```

Wat is het antwoord van de Prolog interpreter bij de volgende queries (inclusief backtracken)?

```
?- p(X), q(X,Y), r(Y,Z), !.  
?- p(X), !, q(X,Y), r(Y,Z).
```

Vraag 4: programma (15 punten)

Schrijf een programma voor het predicaat "aangrenzend", met de volgende specificatie:

aangrenzend(X,Y,Zs) is waar als de elementen X en Y aangrenzend voorkomen in de lijst Zs.
Enkele voorbeelden:

?- aangrenzend(2,3,[1,2,3,4]).
Yes

?- aangrenzend(1,3,[1,2,3,4]).
No

?- aangrenzend(X,Y,[1,2,3,4]).

X = 1
Y = 2 ;

X = 2
Y = 3 ;

X = 3
Y = 4 ;
No

Vraag 5: Definite clause grammars (DCG's) (20 punten)

Vraag 5a

In het doolhof kun je alleen stappen vooruit (v) , achteruit (a), linksaf (l) en rechtsaf (r) maken. Echter je kunt alleen rechtsaf nadat je vorige stap vooruit was. Verder moet je altijd een veelvoud van 2 (2,4,..) naar rechts gaan voordat je weer een andere richting uit kunt.

Voorbeelden van een goed pad zijn: "v v l a v r r a" en "v r r r r l".

In Prolog:

```
?- doolhofpad([v,v,l,a,v,r,r,a], []).  
Yes  
?- doolhofpad([v,r,r,r,r,l], []).  
Yes
```

Voorbeelden van een fout pad zijn: "v l r r" en "v r r r a".

In Prolog:

```
?- doolhofpad([v,l,r,r], []).  
No  
?- doolhofpad([v,r,r,r,a], []).  
No
```

Vraag: Geef met behulp van DCG's een beschrijving van een pad in een doolhof.

Vraag 5b

Deze vraag gaat over de semantiek van een zin. Neem als semantiek van een pad de eindpositie waar je uiteindelijk terecht komt. Begin te tellen vanaf coördinaat (0,0). Een stap naar rechts telt bij de x-coördinaat 1 op, een stap naar links trekt bij de x-coördinaat 1 af, een stap naar voren telt bij de y-coördinaat 1 op, een stap naar achter trekt bij de y-coördinaat er 1 af. De betekenis van de zin "v r r a" is dus (0,2).

De semantiek van een zin kan je op twee manieren bepalen:

1. In twee stappen: De semantiek van een zin (bijv. van "v r r a") kun je bepalen door twee stappen. De eerste stap is het opbouwen van een parse-tree tijdens het parseren van de zin. Je geeft de parse-tree dan als argument terug (dus *doolhofpad(-ParseTree,[v,v,l,a,v,r,r,a],[])*). De tweede stap is het interpreteren van de parse-tree (dus *meaning(+ParseTree,-Meaning)*). Voor het verkrijgen van de betekenis van "v r r a" moet je dan de volgende query stellen:

```
?- doolhofpad(ParseTree,[v,r,r,a],[ ]),meaning(ParseTree,Meaning).
```

2. In één stap: Zoals je weet kun je ook in de DCG's Prolog clauses tussen accoladen opnemen. Het wordt daardoor mogelijk om de twee stappen (parse-tree opbouwen en betekenis bepalen van de parse-tree) in 1 stap te doen. De betekenis van de zin moet je dan dus als extra argument teruggeven.

Voorbeeld:

```
?- doolhofpad(Meaning,[v,r,r,a],[ ]).  
Meaning = (0,2)
```

Vraag: Neem de DCG's uit opgave (a) als uitgangspunt en breid je programma zodanig uit dat de betekenis een pad bepaald wordt. Je mag zelf weten op welke manier je de betekenis bepaalt (zie hierboven beschreven).

Einde tentamen.