

Prolog Toets 2: 19 April 1999

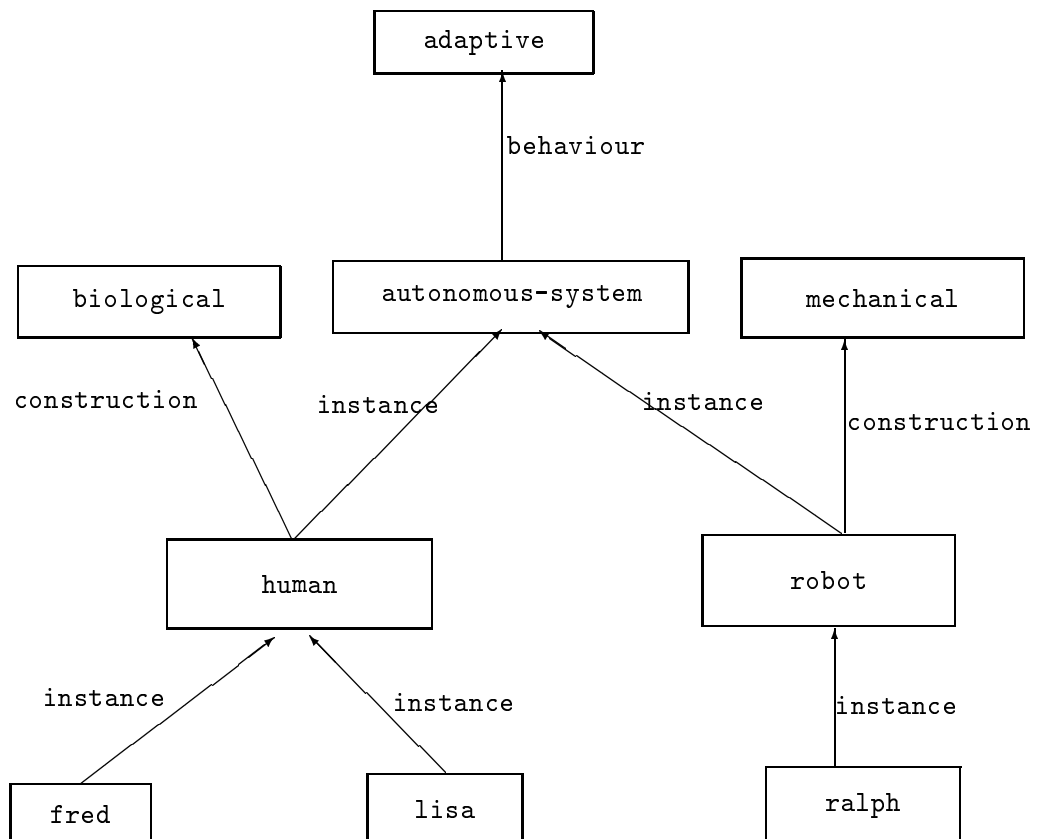
<i>vraag</i>	<i>aantalpunten</i>
1 Semantisch netwerk	30($a : 10, b : 5, c : 5, d : 10$)
2 Ingebouwde predicaten	15
3 Grammatica	25($a : 15, b : 10$)
4 Retract	20($a : 5, b : 15$)

Cijfer = (aantal punten + 10)/100

Veel succes!

Vraag 1: Semantische netwerken (25 punten)

Gegeven het onderstaande semantische netwerk.



1a: (10 punten)

Geef een representatie in Prolog voor het bovenstaande semantische netwerk.

Aan de feiten uit vraag 1a, 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) .
```

1b: (5 punten)

Welke vraag moet je nu aan de prolog interpreter stellen om te achterhalen wat het “behaviour” is van “ralph”?

1c: (5 punten)

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

1d: (10 punten)

Geef een voorbeeld van een query die slaagt door de tweede clause van “fact”. Geef ook de afleidingsboom van deze query.

Gebruik weer voor het voorbeeld het gegeven semantische netwerk.

2: Prolog built-in procedures (15 punten)

Wat is het antwoord van de Prolog interpreter?

1. `?- foo(a,b,c) =.. X.`
2. `?- X =.. [b,Y,[1,2]].`
3. `?- X = ..[X1,X2].`
4. `?- A = B.`
5. `?- A == B.`
6. `?- A is B.`
7. `?- A := B.`
8. `?- X = Y, Y==X.`

Gegeven de volgende feiten:

```
class(volkskrant, krant).
class(nrc, krant).
class(telegraaf, krant).
class(de_tijd, tijdschrift).
class(hp, tijdschrift).
class(geografic, tijdschrift).
class(op_pad, tijdschrift).
```

Wat is het antwoord van de Prolog interpreter (met backtracken, dus ingeven van “;”) op de volgende queries?

9. `?- bagof(Leesvoer, class(Leesvoer,krant),Result).`
10. `?- bagof(Leesvoer, class(Leesvoer,Class), Result).`
11. `?- setof(Class/Leesvoer, class(Leesvoer,Class), Result).`
12. `?- findall(Leesvoer,class(Leesvoer,Class),Result).`
13. `?- findall(Leesvoer,class(Leesvoer,strip),Result).`
14. `?- setof(Leesvoer,class(Leesvoer,strip),Result).`
15. `?- setof(Leesvoer,Class^class(Leesvoer,Class),Result).`

Vraag 3: Definite clause grammars (DCG's) (25 punten)

3a: (15 punten)

Schrijf met behulp van DCG's een grammatica voor het beschrijven van een pad in een doolhof. 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
```

3b: Parse-tree (10 punten)

Geef de parse-tree van de zin "l a v r r a". Geef daarbij aan wat terminals en nonterminals zijn door de terminals te onderstreep en de non-terminals niet.

vraag 4: Retract (20 punten)

Gegeven de volgende clauses:

```
p(1).  
p(2).  
p(3).  
p(X) :- q(X).  
q(1).  
q(a).
```

4a: (5 punten)

Wat is het antwoord van de Prolog interpreter (met eventueel backtracken) bij de query:

```
?- retract(p(1)).
```

Wat is *vervolgens* het antwoord van de Prolog interpreter (met backtracken) op:

```
?- p(X).
```

Wat is *vervolgens* het antwoord van de Prolog interpreter (met backtracken) op:

```
?- retract(p(X)).
```

Wat is *vervolgens* het antwoord van de Prolog interpreter (met backtracken) op:

```
?- p(X).
```

4b: 15 punten

Schrijf een predicaat “retractall(X)”. Dit predicaat moet alle *feiten* en *clauses met head X* uit de prolog database verwijderen.

Gegeven de volgende clauses:

```
p(1).  
p(2).  
p(3).  
p(X) :- q(X).  
q(1).  
q(a).
```

“?- retractall(p(X)).” heeft dan als resultaat dat de volgende clauses overblijven:

```
q(1).  
q(a).
```