

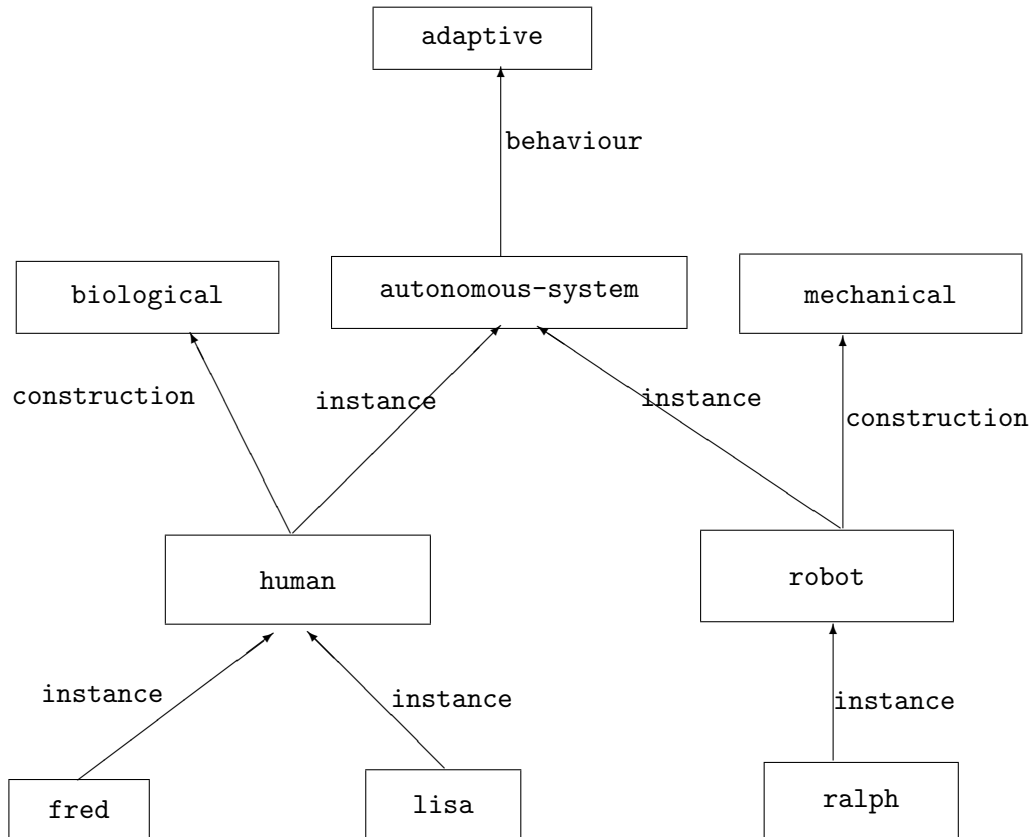
# Tentamen Programmeren in Prolog: 16 augustus 2004

| <i>VRAAG</i>                    | AANTAL PUNTEN      |
|---------------------------------|--------------------|
| 1. Semantisch netwerk           | 15                 |
| 2. Afleidingsboom, Cut operator | 20                 |
| 3. Lijsten                      | 10                 |
| 4. Grammatica                   | 25                 |
| 5. Programma schrijven          | 20                 |
| <i>Cijfer :</i>                 | $(aantal + 10)/10$ |

**Veel succes!**

## Vraag 1: Semantische netwerken (15 punten)

Gegeven het onderstaande semantische netwerk.



Het semantische netwerk wordt in Prolog gerepresenteerd, en onderstaande procedure wordt gebruikt (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) .
```

**1a: (5 punten)**

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

**1b: (5 punten)**

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

**1c: (5 punten)**

Geef een voorbeeld van een query die slaagt door de tweede clause van “fact”. Licht je antwoord toe.

Gebruik weer voor het voorbeeld het gegeven semantische netwerk.

## Vraag 2: Afleidingsboom (20 punten)

Geef de complete afleidingsboom bij het berekenen van alle antwoorden (dwz bij volledig back-tracken) van de query “?-p (Y),w (Y) .”.

- Geef daarbij aan welke regel gebruikt wordt door het nr. van de regel bij de linken te schrijven.
- Geef aan als een tak uiteindelijk faalt, door fail bij deze tak te zetten.
- Geef duidelijk aan welke paden weggesneden worden door de “!”.

```
(1) p(X):-q(X),r(X),!.
(2) p(foo).
(3) q(a).
(4) q(b).
(5) q(c).
(6) r(b).
(7) r(c).
(8) s(b).
(9) s(c).
(10) w(U):-s(U).
```

### Vraag 3: Lijsten (10 punten)

#### vraag 3a: (5 punten)

Geef de lijst  $[1, 2, 3]$  weer in een boom representatie.

#### vraag 3b (5 punten):

Wat is het antwoord van de prolog-interpreter op onderstaande queries? Geef uiteraard ook de bindingen van de variabelen.

```
?- [1,[b,c],d] = [X|Y].  
?- [a,b|[]] = [a,b].  
?- [a,b|[c,d]] = [X|Y].  
?- [f(X),X,f(Y)] = [f(1)|Z].  
?- [1,3,5] = [X,Y]
```

### Vraag 4: Programma (20 punten)

“maplist” is een ingebouwd predicaat. Het heeft de volgende definitie:

```
maplist(+Pred, ?List1, ?List2)  
    Pas het predicaat Pred toe op alle opeenvolgende paren uit  
    List1 en List2.  
    Faalt als Pred niet toegepast kan worden op een paar.
```

Een voorbeeld van het toepassen van maplist:

```
?- maplist(kwadraat, [0, 1, 2], X).  
  
X = [0, 1, 4]
```

Waarbij de definitie voor kwadraat is:

```
kwadraat(X,Y) :- Y is X*X.
```

Schrijf de code voor de maplist procedure.

## Vraag 5: Definite clause grammars (DCG's) (25 punten)

### 5a: (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
```

### 5b: 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.

**Einde tentamen.**