

Tentamen Programmeren in Prolog: 27 mei 2004

<i>VRAAG</i>	AANTAL PUNTEN
1.Unificatie	15
2.Afleidingsboom & cut	25
3.Programma schrijven	25
4.Grammatica	25
<i>Cijfer :</i>	$(aantal + 10)/10$

Veel succes!

Vraag 1: Unificatie (15 punten)

Geef de antwoorden die de Prolog interpreter zou geven (inclusief de eventuele bindingen van de variabelen).

1. ?- $[a,b|X]=[A,B,c]$.
2. ?- $[a,b]=[b,a]$.
3. ?- $[a|[b,c]]=[a,b,c]$.
4. ?- $[a,[b,c]]=[a,b,c]$.
5. ?- $[a,X]=[X,b]$.
6. ?- $[a|[]]=[X]$.
7. ?- $[a,b,X,c]=[A,B,Y]$.
8. ?- $[H|T]=[[a,b],[c,d]]$.
9. ?- $[[X],Y]=[a,b]$.
10. ?- $[X|Y]=[a]$.
11. ?- $X = .. [b,Y,[1,2]]$.
12. ?- $X = .. [X1,X2]$.
13. ?- $X \text{ is } 1, Y \text{ is } X+1, Y < 3$.
14. ?- $Y < 3, X \text{ is } 1, Y \text{ is } X+1$.
15. ?- $X=3, Y=X+1, Z \text{ is } Y$.

Vraag 2: Afleidingsboom (25 punten)

Gegeven onderstaand programma:

```
(1) b(1).  
(2) b(11).  
(3) c(2).  
(4) c(22).  
(5) a(1,1).  
(6) a(X,Y) :- b(X), !, c(Y).  
(7) a(0,0).
```

Vraag 2a (15)

Geef de complete afleidingsboom bij het berekenen van alle antwoorden (dwz bij volledig backtracken) van de query “?- a(X,Y) .”.

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

Vraag 2b (10)

Wat zijn alle antwoorden (dwz. bij volledig backtracken) die de Prolog interpreter geeft op de query “?- a(X,Y), !, b(B)”. Geef de antwoorden in de volgorde waarop de Prolog interpreter ze afleidt.

Vraag 3: Programma schrijven (25 punten)

Vraag 3a (15)

Schrijf een programma dat het verschil tussen twee sets bepaalt ($Set1 \setminus Set2$).

set_verschil(+Set1, +Set2, −Set3): *Set3* is de verzameling die het verschil weergeeft tussen de verzamelingen Set1 en Set2. Een verzameling is gerepresenteerd als een lijst.

Voorbeeld:

```
?- set_verschil([1,2,3,4],[3,4,5,6],Verschil).  
Verschil = [1,2];  
No.
```

```
?- set_verschil([2,1,3,4],[3,4,5,6],Verschil).  
Verschil = [2,1];  
No
```

Vraag 3b (10)

Is het mogelijk je programma (uit vraag 3a) te optimaliseren door gebruik te maken van een groene cut? Geef een toelichting op je antwoord.

Vraag 4: Grammatica (25)

Gegeven zijn de volgende DCG's:

```
expr --> basicexpr, addterm.  
expr --> basicexpr, factorterm.  
expr --> basicexpr.  
basicexpr --> [1].  
basicexpr --> [2].  
basicexpr --> [3].  
addterm --> [plus], expr.  
factorterm --> [maal], expr.
```

Deze DCG's kunnen gebruikt worden voor het genereren en testen van bepaalde expressies, namelijk een expressie bestaande uit de elementen "1", "2", "3", "maal", "plus". De plus en de maal zijn hierbij infix operatoren. Een valide expressie is bijv. "1 plus 2", een niet valide expressie is "1 1 plus".

Enkele voorbeelden van het gedrag:

```
?- expr([1, plus, 2, maal, 3], []).  
Yes.
```

```
?- expr([1], []).  
Yes.
```

```
% alleen de getallen 1,2 en 3 zijn valide getallen.  
?- expr([4], []).  
No.
```

```
?- expr([1, 1], []).  
No.
```

```
?- expr([plus, 1, maal, 3], []).  
No.
```

Vraag 4a (15)

Breid bovenstaande DCG regels uit zodat er een parse tree opgebouwd wordt. Hiervoor kan je een nieuw argument toevoegen aan een DCG regel. Gebruik in de parse tree de structuren `maal(...)`, en `plus(...)`, en verder de getallen 1,2, en 3. Je kan ervan uitgaan dat "plus" en "maal" gelijke precedenties hebben, en dat de input simpel van links naar rechts geevalueerd wordt. "2 maal 2 plus 3" heeft de bijbehorende parse tree `maal(2, plus(2,3))`.

Voorbeeld:

```
?- expr(Tree,[1, plus, 2, maal, 3],[ ]).
Tree = plus(1, maal(2, 3));
Yes.
```

Vraag 4b (10)

Ga er van uit dat de parse tree bestaat uit de volgende structuren en terminals: maal(...), plus(...), 1, 2, 3. (zoals bij vraag 4a).

Schrijf een programma "meaning" dat de waarde van de expressie uitrekent, gegeven de parse tree. (meaning(+ParseTree,-Meaning)). Je kan ervan uitgaan dat "plus" en "maal" gelijke precedenties hebben, en dat de input simpel van links naar rechts geevalueerd wordt. Dus "2 maal 2 plus 3" heeft de parse tree maal(2, plus(2,3)) en dus de waarde $2 * (2 + 3) = 10$.

Voorbeeld:

```
?- expr(Tree,[1, plus, 2, maal, 3],[ ]), meaning(Tree,Value).
Tree = plus(1, maal(2, 3)),
Value = 7;
No.
```

Einde tentamen.