

Prolog hertentamen: 13 augustus 1999

VRAAG	AANTAL PUNTEN
1.Lijsten	15(10 + 5)
2.Geven van een afleidingsboom	15
3.Fail operator	10
4.Programma schrijven	20
5.Grammatica	20
6.Assert en cut	10

Cijfer = (aantal punten + 10) / 10

Veel succes!

Unificaties (15 punten)

A. (10 punten)

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

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. ?- $[X | Y] = [a]$.

\ANTWOORD{

1. ?- $[a, b | X] = [A, B, c]$.

X = [c]

A = a

B = b ;

No

$$2. \quad ?- \quad [a, b] = [b, a] .$$

No

$$3. \quad ?- \quad [a \mid [b, c]] = [a, b, c] .$$

Yes

$$4. \quad ?- \quad [a, [b, c]] = [a, b, c] .$$

No

$$5. \quad ?- \quad [a, X] = [X, b] .$$

No

$$6. \quad ?- \quad [a \mid []] = [X] .$$

$$X = a ;$$

No

$$7. \quad ?- \quad [a, b, X, c] = [A, B, Y] .$$

No

$$8. \quad ?- \quad [H \mid T] = [[a, b], [c, d]] .$$

$$H = [a, b]$$

$$T = [[c, d]] ;$$

No

$$9. \quad ?- \quad [[X], Y] = [a, b] .$$

No

$$10. \quad ?- \quad [X \mid Y] = [a] .$$

$$X = a$$

$$Y = [] ;$$

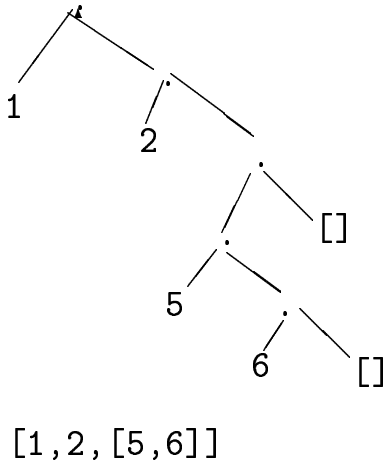
No

}

B. (5 punten)

Geef de boomrepresentatie van de lijst $[1, 2, [5, 6]]$.

Antwoord:



Vraag 2: Afleidingsboom (15 punten)

Geef de complete afleidingsboom bij het berekenen van alle antwoorden (dwz bij volledig backtracken) 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) $p(\text{foo}).$

(3) $q(a).$

(4) $q(b).$

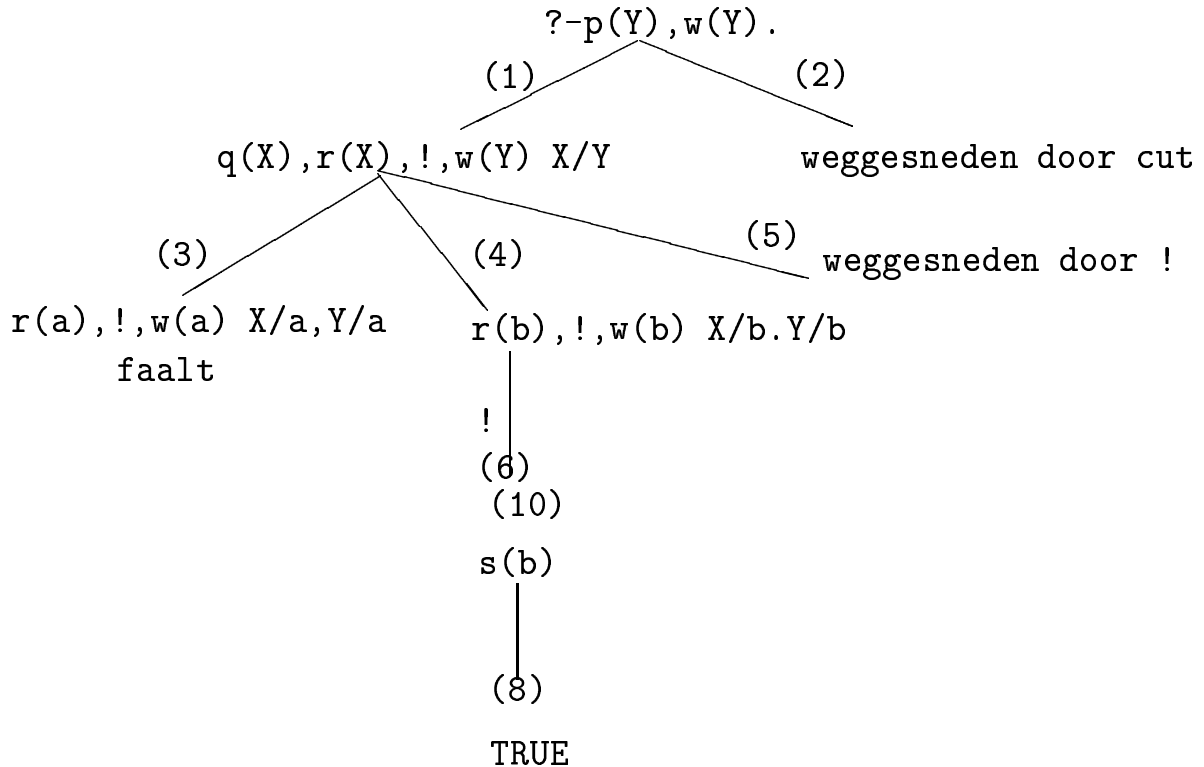
(5) $q(c).$

(6) $r(b).$

(7) $r(c).$

(8) `s(b).`
 (9) `s(c).`
 (10) `w(U):-s(U).`

Antwoord:



Vraag 3: fail operator (10 punten)

Definieer het systeem predicaat `\==` door gebruik te maken van `==` en *fail*. Het is niet toegestaan om het systeem predicaat

`\+`

te gebruiken. Noem het predicaat dat je gaat schrijven *ongelijk*.

```

\ANTWOORD{
ongelijk(X,Y) :- X==Y, !, fail.
ongelijk(X,Y).
}

```

Vraag 4: Programma schrijven (20 punten)

- a: Schrijf een predicaat `prefix(Prefix,List)` dat slaagt als `Prefix` de prefix van de lijst `List` is. Voorbeeld: `prefix([1,2],[1,2,3])` slaagt, en `prefix([2],[1,2])` faalt.
- b: Schrijf een predicaat `suffix(Suffix,List)` dat slaagt als `Suffix` de suffix van de lijst `List` is. Voorbeeld: `suffix([2,3],[1,2,3])` slaagt, en `suffix([1],[1,2])` faalt.
- c: Schrijf een predicaat `sublist1(List1,List2)` dat slaagt als `List1` een sublist is van `List2`. Gebruik hierbij zowel het prefix als het suffix predicaat. Voorbeeld: `sublist1([2,3],[1,2,3])` slaagt, en `sublist1([4,3],[1,2,3])` faalt.
- d: Schrijf een predicaat `sublist2(List1,List2)` dat slaagt als `List1` een sublist is van `List2`. Gebruik hierbij wel het prefix predicaat, maar niet het suffix predicaat.

\ANTWOORD{

```
prefix([],Ys).  
prefix([X|Xs],[X|Ys]):- prefix(Xs,Ys).
```

```
suffix(Xs,Xs).  
suffix(Xs,[Y|Ys]):- suffix(Xs,Ys).
```

```
sublist1(Xs,Ys):- prefix(Ps,Ys), suffix(Xs,Ps).
```

```
sublist2(Xs,Ys):- prefix(Xs,Ys).  
sublist2(Xs,[Y|Ys]):-sublist(Xs,Ys).  
}
```

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

Schrijf een programma met behulp van DCG's die expressies van de vorm $a^*b^*c^*$ herkent. a^* betekent dat a 0 of meer keren voorkomt. Voorbeelden

van deze expressies zijn dus: bc , aaa , $abcc$, etc. De semantiek van de expressie is een waarde: a telt voor 1, b telt voor 2, en c telt voor 3.

“?- s(N,[a,a,b,b,b,c],[]).” levert dus “N=11” op ($1+1+2+2+2+3=11$).

\ANTWOORD{

s(N) --> a(Na), b(Nb), c(Nc), {N is Na+Nb+Nc}.

a(N) --> [a], a(N1), {N is N1+1}.

a(0) --> [].

b(N) --> [b], b(N1), {N is N1+2}.

b(0) --> [].

c(N) --> [c], c(N1), {N is N1+3}.

c(0) --> [].

}

Vraag 6: Assert en cut (10 punten)

Gegeven onderstaand programma “assert-cut”.

p1(1).

p1(2).

p2(3).

p2(4).

p3(5).

p3(6).

p11(1).

p11(2).

p22(3).

p22(4).

p33(5).

p33(6).

p1(X):-assert(r(X)),!,p1(X),!.

p3(X):-assert(t(X)),p3(X).

p22(X):-p22(X),!,assert(ss(X)).

p33(X):-p33(X),assert(tt(X)).

Onderstaande queries worden in deze volgorde aan de interpreter aangeboden. Geef het antwoord dat de interpreter geeft als je een “;” steeds intikt.

```
?- p1(X) .  
?- p3(X) .  
?- r(X) .  
?- t(X) .  
?- p22(X) .  
?- p33(X) .  
?- ss(X) .  
?- tt(X) .  
?- retract(p33(X)) .  
?- p33(X) .
```

```
\ANTWOORD{
```

```
?- p1(X) .
```

```
X = 1 ;
```

```
X = 2 ;
```

```
X = 1 ;
```

```
No
```

```
?- p3(X) .
```

```
X = 5 ;
```

```
X = 6 ;
```

```
X = 5 ;
```

```
X = 6 ;
```

X = 5 ;

X = 6 ;

X = 5 ;

X = 6 ;

X = 5 ;

X = 6 ;

X = 5 ;

X = 6 ;

X = 5

Yes

?- r(X) .

X = _G93 ;

No

?- t(X) .

X = _G93 ;

X = _G93 ;

X = _G93 ;

X = _G93 ;

X = _G93 ;

$X = _G93$;

No

?- p22(X) .

$X = 3$;

$X = 4$;

$X = 3$;

No

?- p33(X) .

$X = 5$;

$X = 6$;

$X = 5$;

$X = 6$;

$X = 5$;

$X = 6$;

$X = 5$;

$X = 6$;

$X = 5$;

$X = 6$;

$X = 5$;

$X = 6$;

$X = 5$;

$X = 6$;

$X = 5$;

$X = 6$;

$X = 5$;

$X = 6$;

$X = 5$

Yes

?- ss(X) .

$X = 3$;

No

?- tt(X) .

$X = 5$;

$X = 6$;

$X = 5$;

$X = 5$;

$X = 6$;

X = 6 ;

X = 5 ;

X = 5 ;

X = 5 ;

X = 6 ;

X = 6 ;

X = 6 ;

X = 5 ;

X = 5 ;

X = 5 ;

X = 5

Yes

?- retract(p33(X)).

X = 5 ;

X = 6 ;

No

?- p33(X).

[WARNING: Out of local stack]

Execution Aborted

}