

Prolog Tentamen: 1 juni 1999

Unifications (15 punten)

?- [X,Y,Z] = [john,likes,fish].

X = john
Y = likes
Z = fish ;

No

?- [cat] = [X|Y].

X = cat
Y = [] ;

No

?- [X,Y|Z]=[mary,likes,wine].

X = mary
Y = likes
Z = [wine] ;

No

?- [[the,Y]|Z] = [[X,hare],[is,here]].

Y = hare
Z = [[is, here]]
X = the ;

No

?- [X,Y|Z,W] = [1,2,3,4].

No

?- [golden|T]=[golden,norfolk].

```

T = [norfolk] ;

No
?- [vale,horse]=[horse,X].

No
?- [white|Q]=[P|horse].

Q = horse
P = white ;

No
?- a(b,C,d(e,F,g(h,i,J)))=a(F,c,d(J,F,g(h,i,J))).

C = c
F = b
J = e ;

No
?- f(1,2)=X(Y,Z).

[WARNING: Syntax error: Operator expected
f(1,2)=X(Y,Z
** here **
) . ]
?- Y = 3, X is 3 + Y.

Y = 3
X = 6 ;

No
?- X is 3 + Y, Y = 3.
[WARNING: Arguments are not sufficiently instantiated]

No
?- X is 12, X == 6 + 6.

X = 12 ;

No

```

?- $X = 12, X = 6 + 6.$

No

?- $f(A,B) == f(X,Y).$

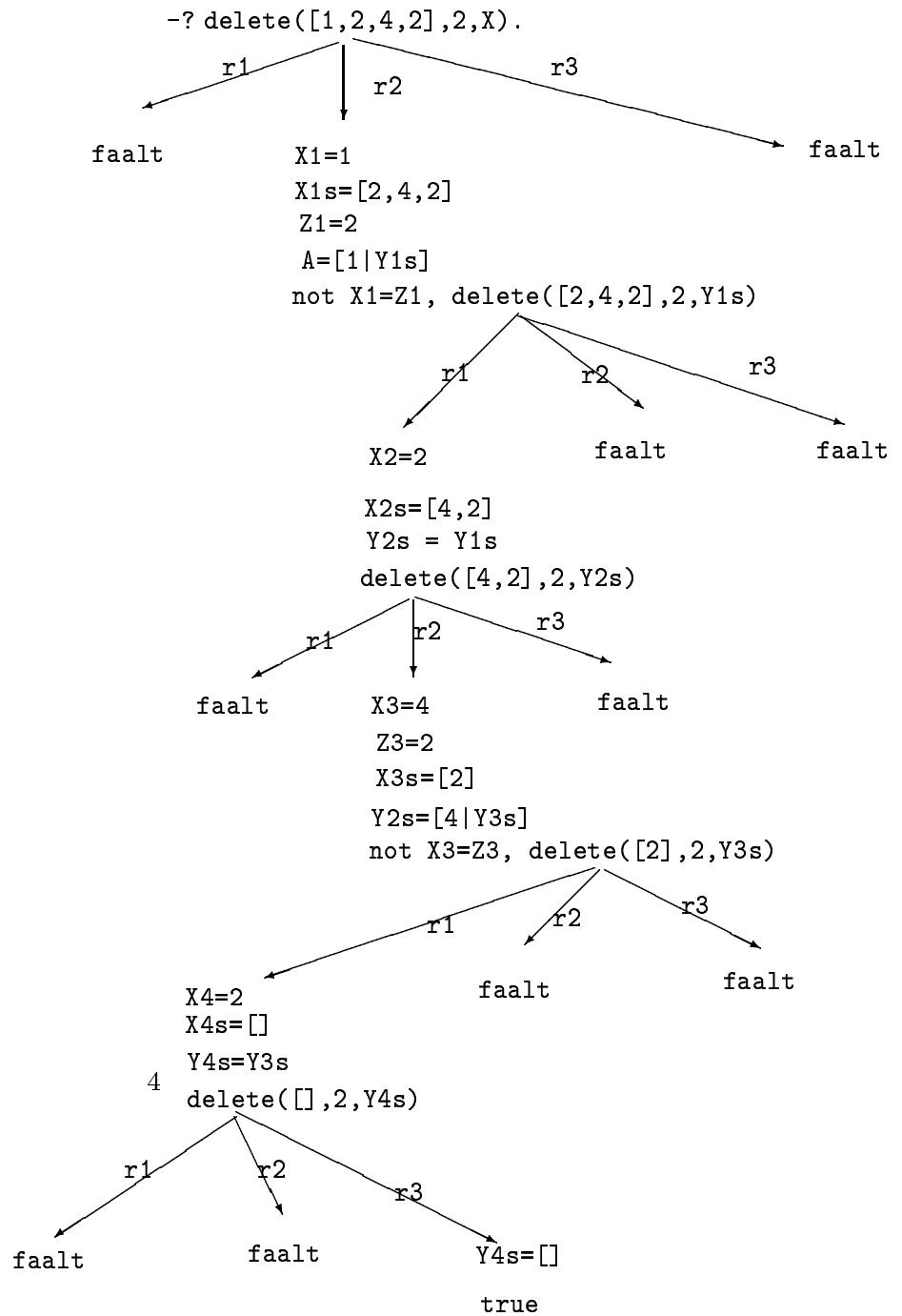
No

Vraag 2: Afleidingsboom (15 punten)

```

r1: delete([X|Xs],X,Ys) :- delete(Xs,X,Ys).
r2: delete([X|Xs],Z,[X|Ys]) :- not X = Z, delete(Xs,Z,Ys).
r3: delete([],X,[]).

```



Vraag 3: Cut operator (20 punten)

3a: if-then-else (15 punten)

```
if_then_else1(P,Q,_):- P, Q.  
if_then_else1(P,_,R):- not P, R.
```

```
if_then_else2(P,Q,_):- P,!, Q.  
if_then_else2(_,_,R):- R.
```

De ! zorgt er voor dat als P waar is de tweede clause niet meer gedaan wordt. Het is een rode cut. De declaratieve betekenis is van het programma met cut anders dan zonder de cut.

Met de cut: $(P \ \& \ Q) \vee (\sim P \ \& \ R)$

Zonder de cut: $(P \ \& \ Q) \vee R$

3b: (10 punten)

```
foo(1).  
foo(2).  
foo(3).
```

```
?- foo(X), !, foo(Y).
```

```
X = 1  
Y = 1 ;
```

```
X = 1  
Y = 2 ;
```

```
X = 1  
Y = 3 ;
```

```
No
```

Vraag 4: Programma (15 punten)

```
mymaplist(_, [], []).
mymaplist(P, [X|L], [Y|M]):-
    Q =.. [P,X,Y],
    call(Q),
    mymaplist(P,L,M).
```

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

5a: Semantiek van een zin (15 punten)

```
doolhofpad(doolhofpad(Stap)) --> stap(Stap).
doolhofpad(doolhofpad(Stap,Doolhofpad)) -->
    stap(Stap), doolfhofpad(Doolhofpad).
stap(stap(v)) --> [v].
stap(stap(l)) --> [l].
stap(stap(a)) --> [a].
stap(stap(v,Rechts)) --> [v], staprechts(Rechts).
staprechts(staprechts(rr)) --> [r],[r].
staprechts(staprechts(rr,Staprechts)) --> [r],[r], staprechts(Staprechts).

meaning(doolhofpad(Stap),M):-
    meaning(Stap,M).

meaning(doolhofpad(Stap,Doolhofpad),M):-
    meaning(Stap,M1),
    meaning(Doolhofpad,M2),
    addcorr(M1,M2,M).

meaning(stap(v),(1,0)).
meaning(stap(a),(-1,0)).
meaning(stap(l),(0,-1)).
meaning(stap(v,Rechts),M):-
    meaning(stap(v),M1),
    meaning(Rechts,M2),
```

```

    addcorr(M1,M2,M).
meaning(staprechts(rr),(0,2)).
meaning(staprechts(rr,Staprechts),M):-
    meaning(staprechts(rr),M1),
    meaning(Staprechts,M2),
    addcorr(M1,M2,M).

addcorr((X1,X2),(Y1,Y2),(Z1,Z2)):-
    Z1 is X1 + Y1,
    Z2 is X2 + Y2.

```

5b: Semantiek van een zin (10 punten)

```

doolhofpad(M) --> stap(M).
doolhofpad(M) --> stap(M1),doolhofpad(M2), {addcorr(M1,M2,M)}.
stap((1,0)) --> [v].
stap((0,-1)) --> [l].
stap((-1,0)) --> [a].
stap(M) --> [v], staprechts(M1), {addcorr((1,0),M1,M)}.
staprechts((0,2)) --> [r],[r].
staprechts(M) --> [r],[r], staprechts(M1), {addcorr((0,2),M1,M)}.

addcorr((X1,X2),(Y1,Y2),(Z1,Z2)):-
    Z1 is X1 + Y1,
    Z2 is X2 + Y2.

```