

## Serie 1

Gruppe 10

Adnan Alyousfi, 218205332, Informatik  
 Dirk Peglow, Informatik  
 Nils Henrik Seitz, 218205308, Informatik  
 Lorka Trad, Informatik  
 Nico Trebbin, 218204402, Informatik

Aufgabe 1

## 1.C

```

8 % 1.A
9 parent(walter,mirco).      %f-p1
10 parent(walter,jana).      %f-p2
11 parent(mirco,nele).       %f-p3
12 parent(silke,nele).       %f-p4
13 parent(mirco,willi).      %f-p5
14 parent(silke,filip).      %f-p6
15 parent(dorothea,dagmar).  %f-p7
16 parent(dagmar,silke).     %f-p8
17 parent(dagmar,jan).       %f-p9
18
19
20 female(dorothea).         %f-f1
21 female(dagmar).           %f-f2
22 female(jana).             %f-f3
23 female(silke).            %f-f4
24 female(nele).             %f-f5
25
26 male(walter).             %f-m1
27 male(mirco).              %f-m2
28 male(jan).                %f-m3
29 male(filip).              %f-m4
30 male(willi).              %f-m5
31
32
33 % 1.B
34 % helper: for brother, sister, aunt, uncle
35 sibling(X,Y) :- parent(A,X),parent(A,Y),X \= Y.      %r0
36
37 child(X,Y) :- parent(Y,X).                          %r1
38 daughter(X,Y) :- female(X),child(X,Y).              %r2
39 son(X,Y) :- male(X),child(X,Y).                     %r3
40 brother(X,Y) :- male(X),sibling(X,Y).               %r4
41 sister(X,Y) :- female(X),sibling(X,Y).              %r5
42 aunt(X,Y) :- parent(Z,Y),sister(X,Z).              %r6
43 uncle(X,Y) :- parent(Z,Y),brother(X,Z).             %r7
44 grandparent(X,Y) :- parent(Z,Y),parent(Z,Y).       %r8

```

```

SWI-Prolog comes with ABSOLUTELY NO WARRANTY.
It is free software.
Please run ?- license. for legal details.

For online help and background, visit
swi-prolog.org
For built-in help, use ?- help(Topic).
pos(Word).

?- [aufgabe1].
true.

?- child(dagmar,dorothea). %true
true.

?- daughter(dagmar,dorothea). %true
true.

?- son(jan,dagmar). %true
true.

?- brother(filip,nele). %true
true.

?- sister(filip,nele). %false
false.

?- aunt(jana,willi). %true
true.

?- uncle(jan,filip). %true
true.

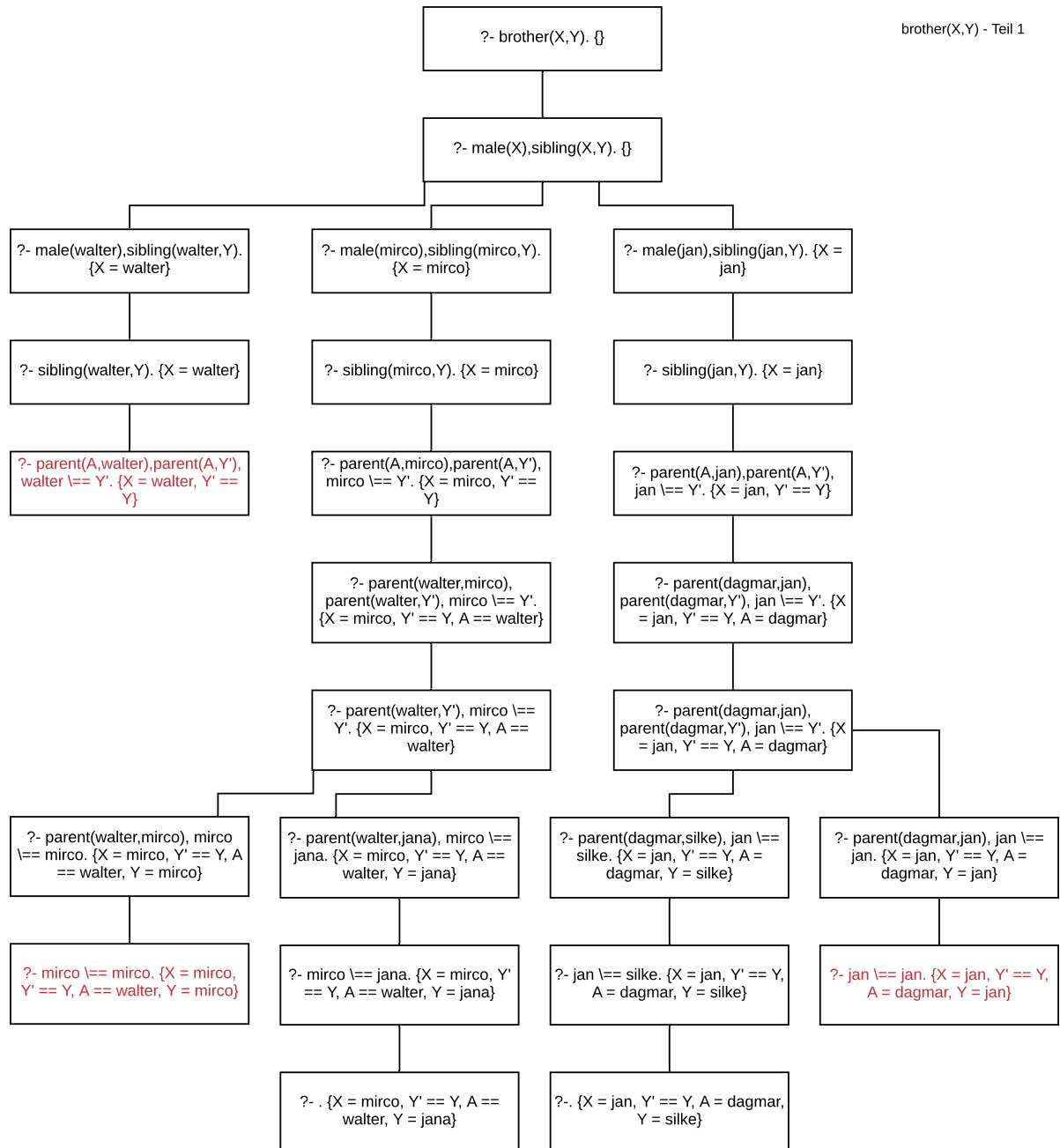
?- grandparent(dorothea,jan). %true
true.

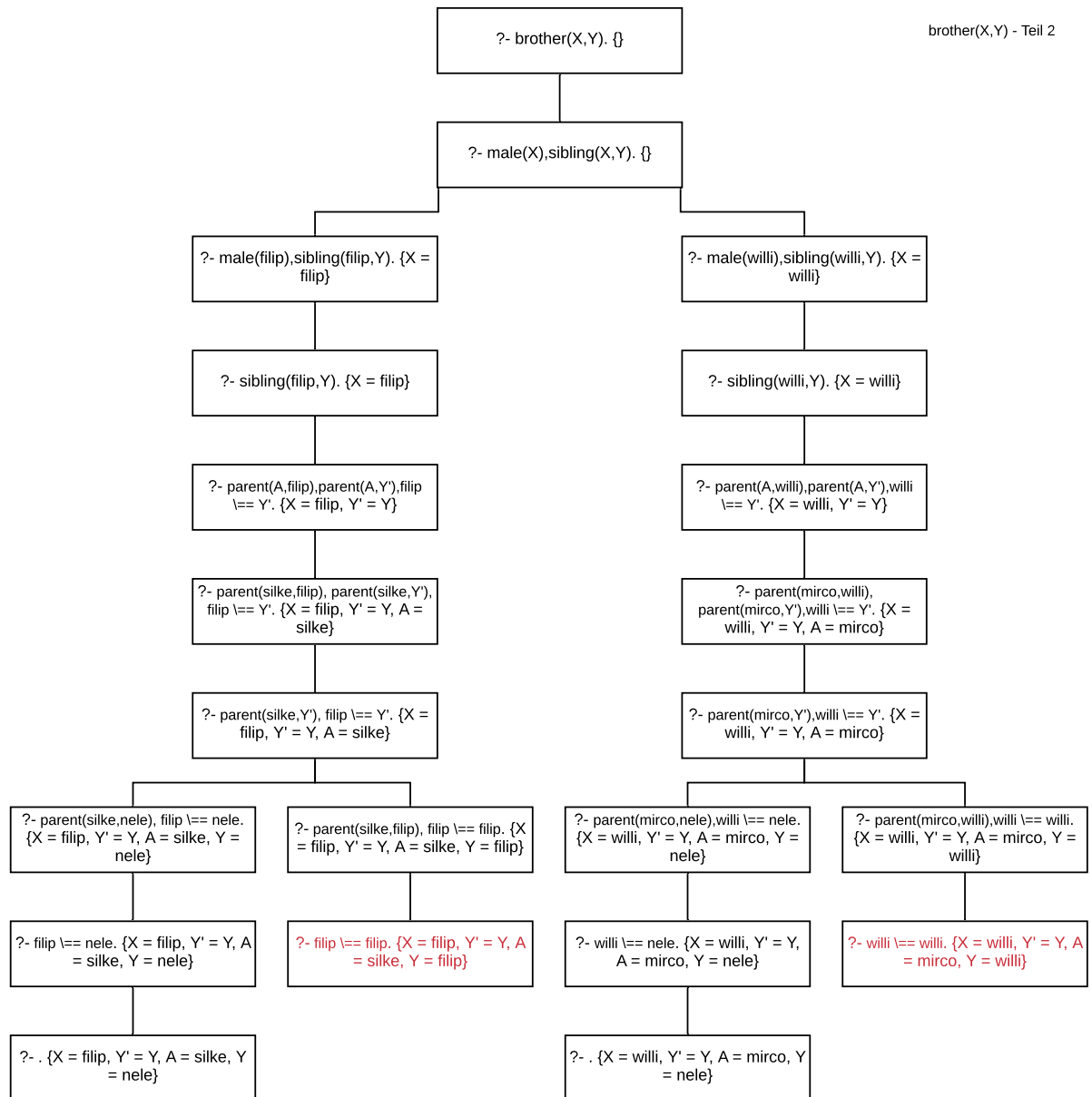
?- halt.
Wilss-Mac:Prolog nils$ 

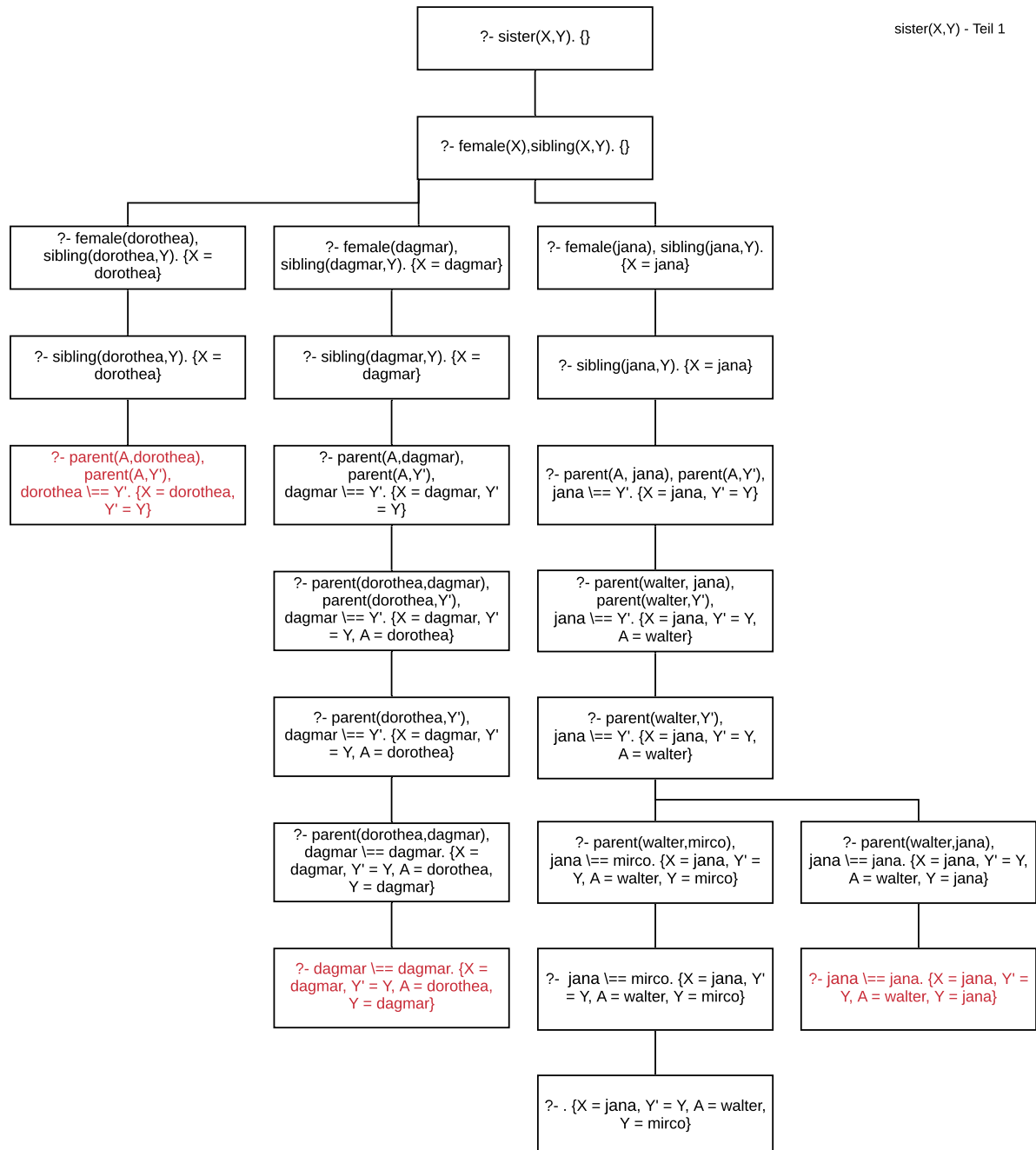
```

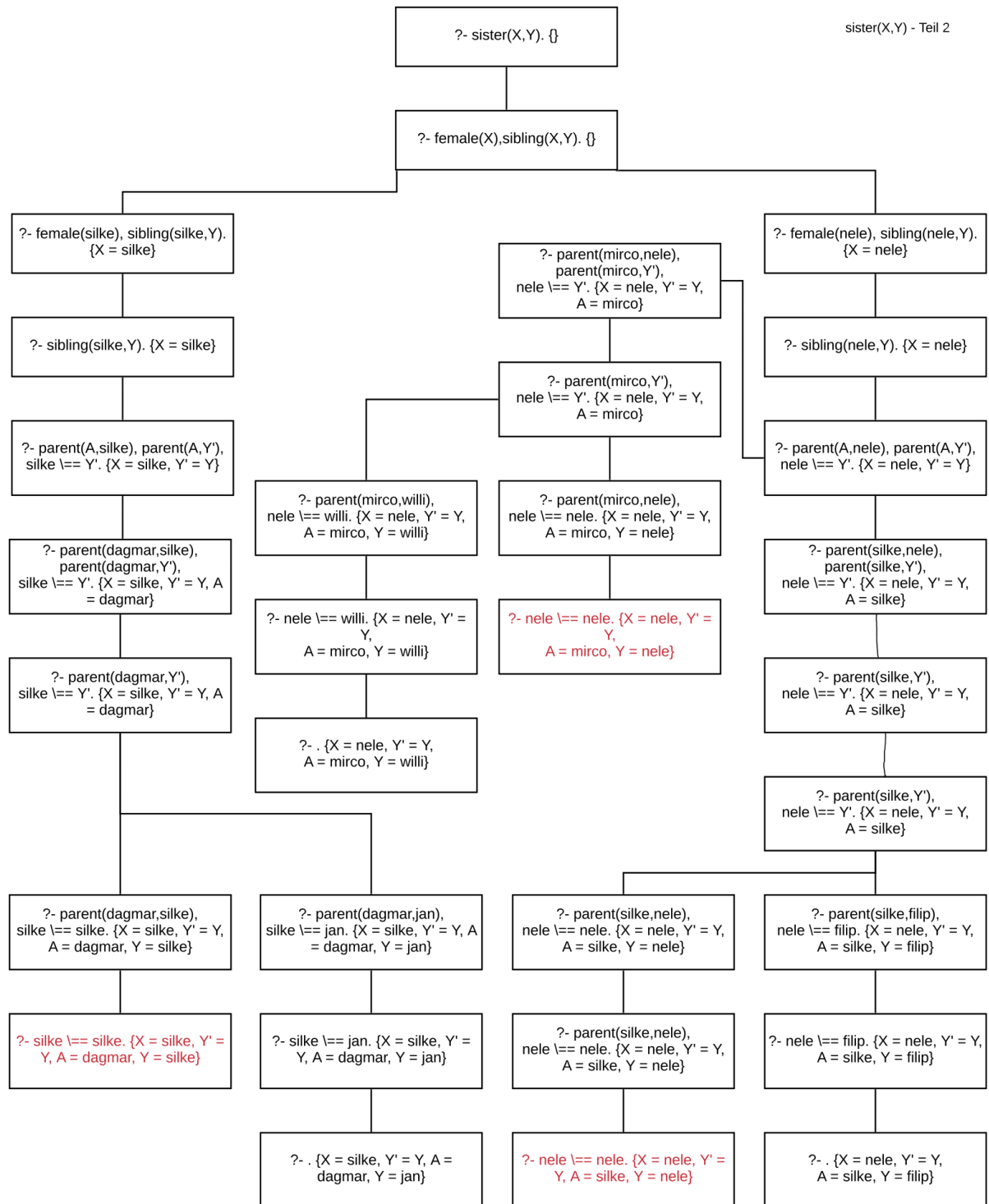
## 1.D

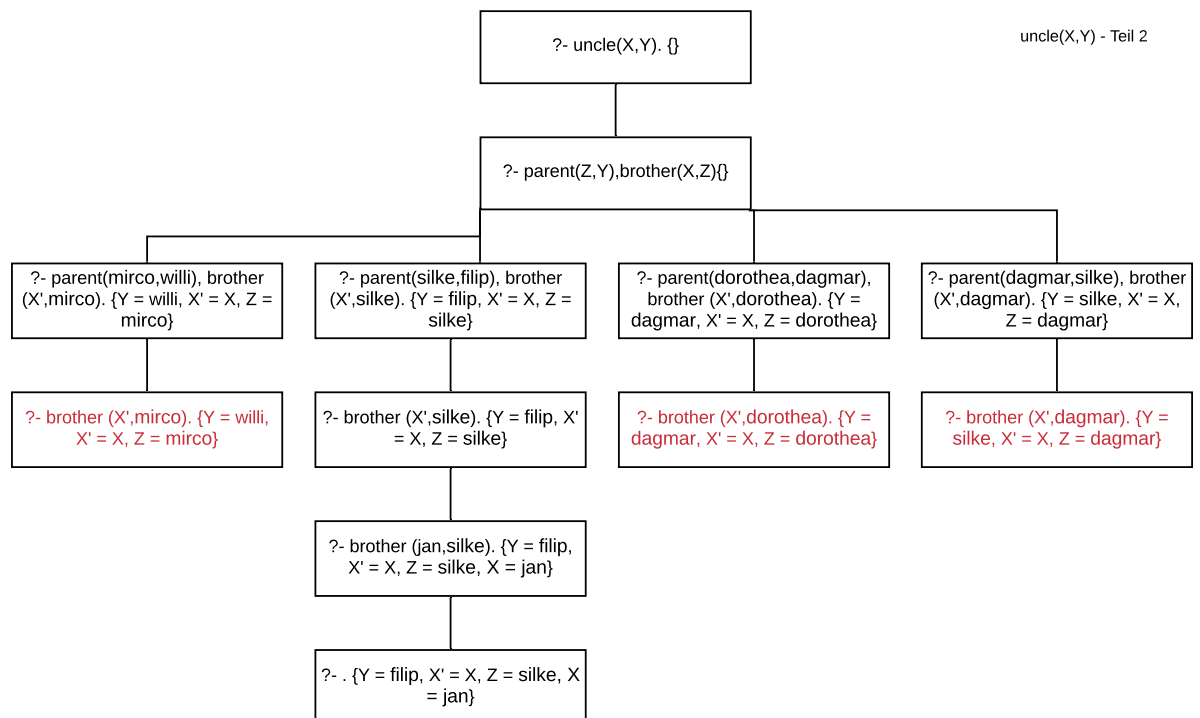
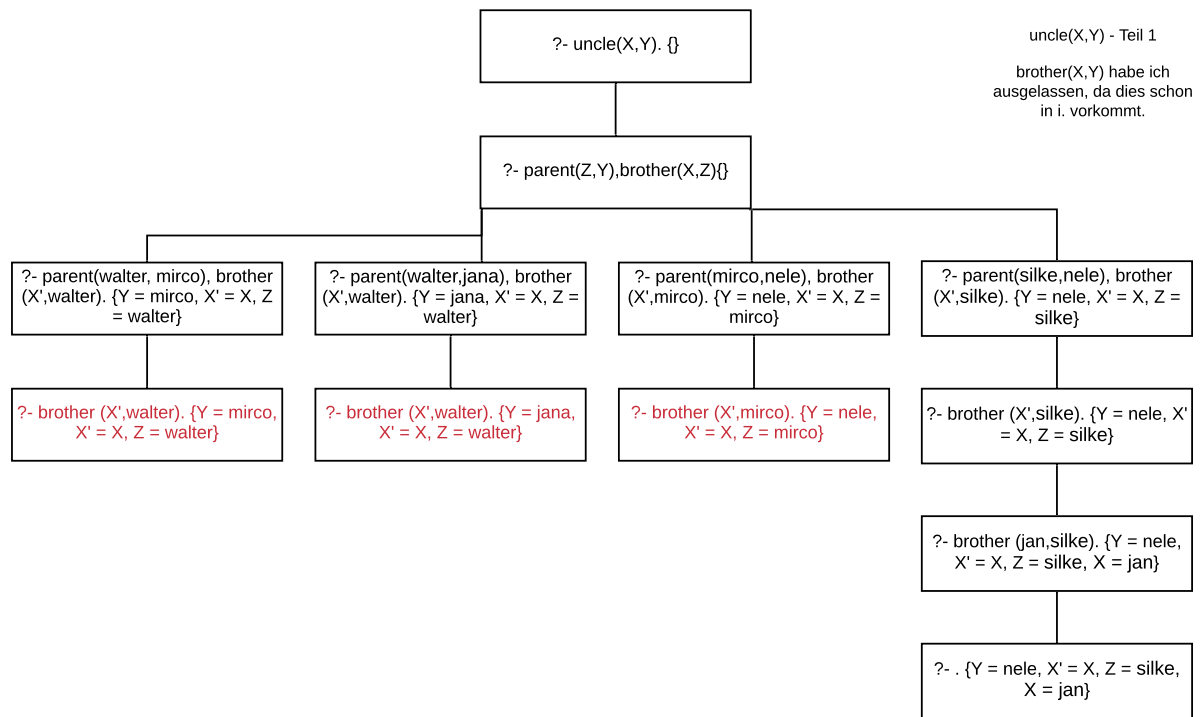
Der Übersichtlichkeit halber habe ich die Ableitungsbäume getrennt aufgeschrieben. Die einzelnen Ableitungsbäume werden der Reihe nach abgelaufen (d.h. Prolog durchläuft zuerst brother(X,Y) - Teil 1, und erst dann Teil 2)



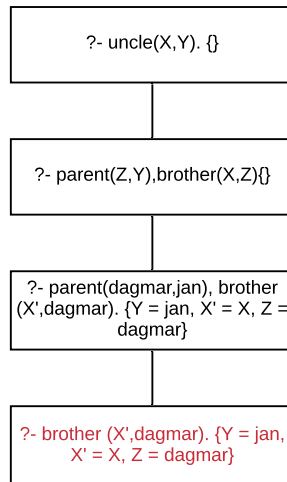


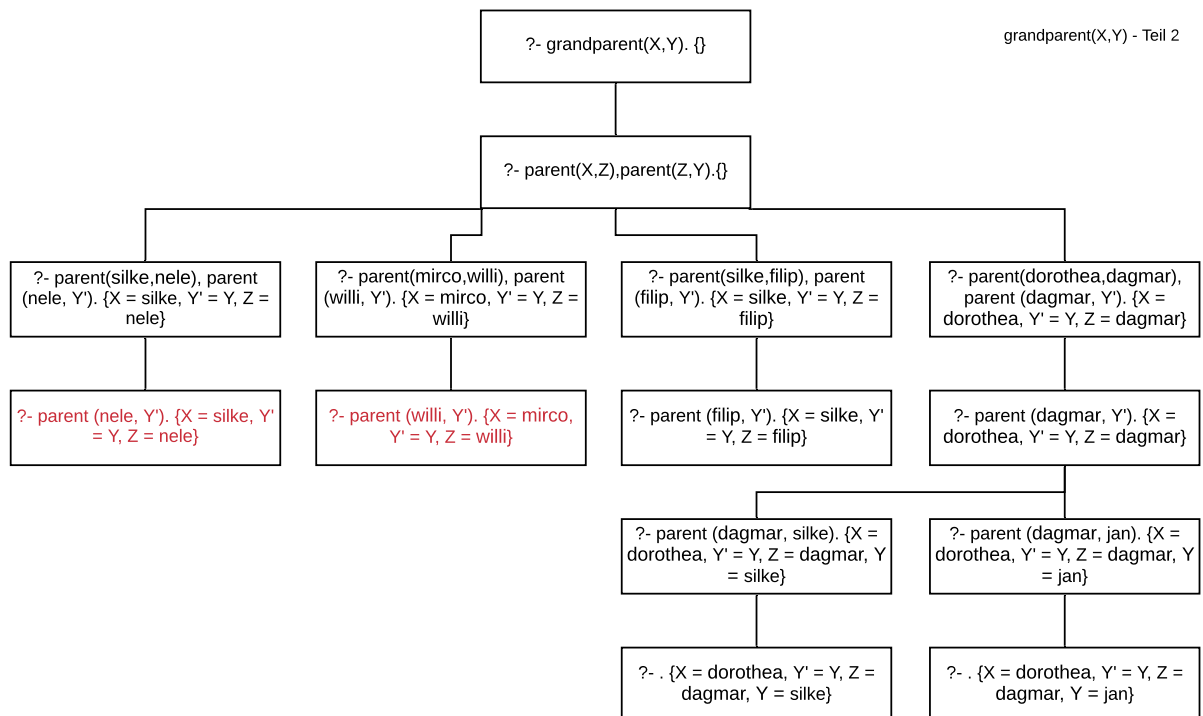
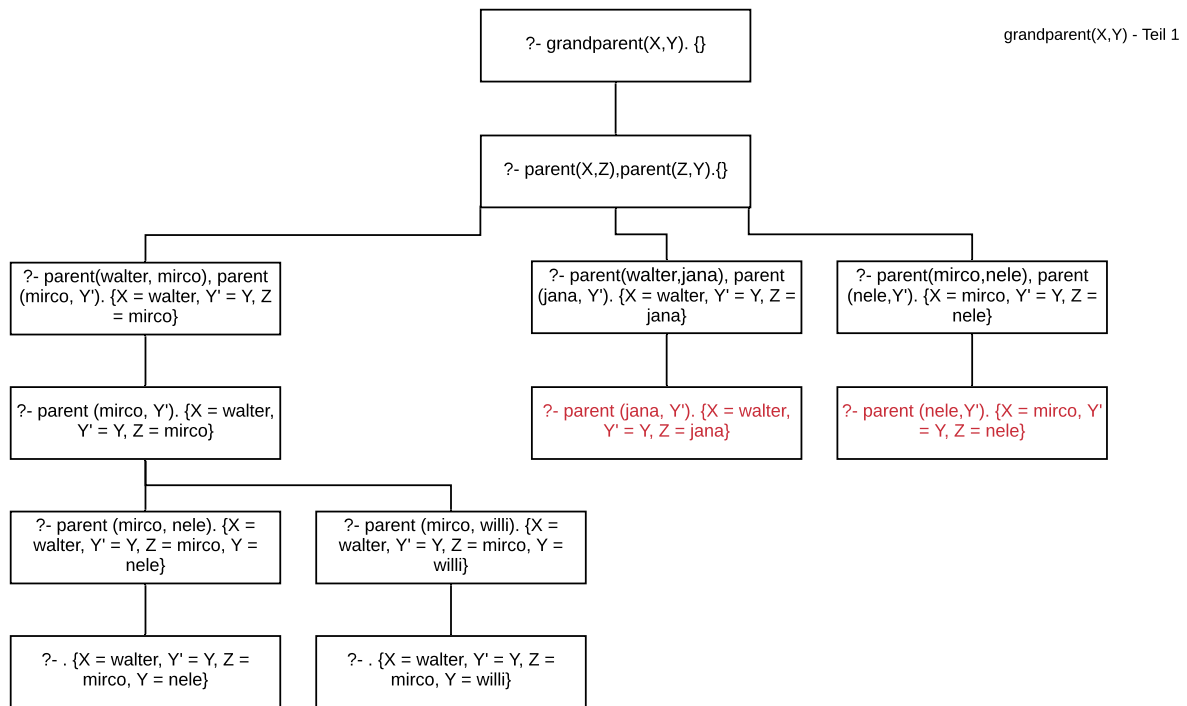




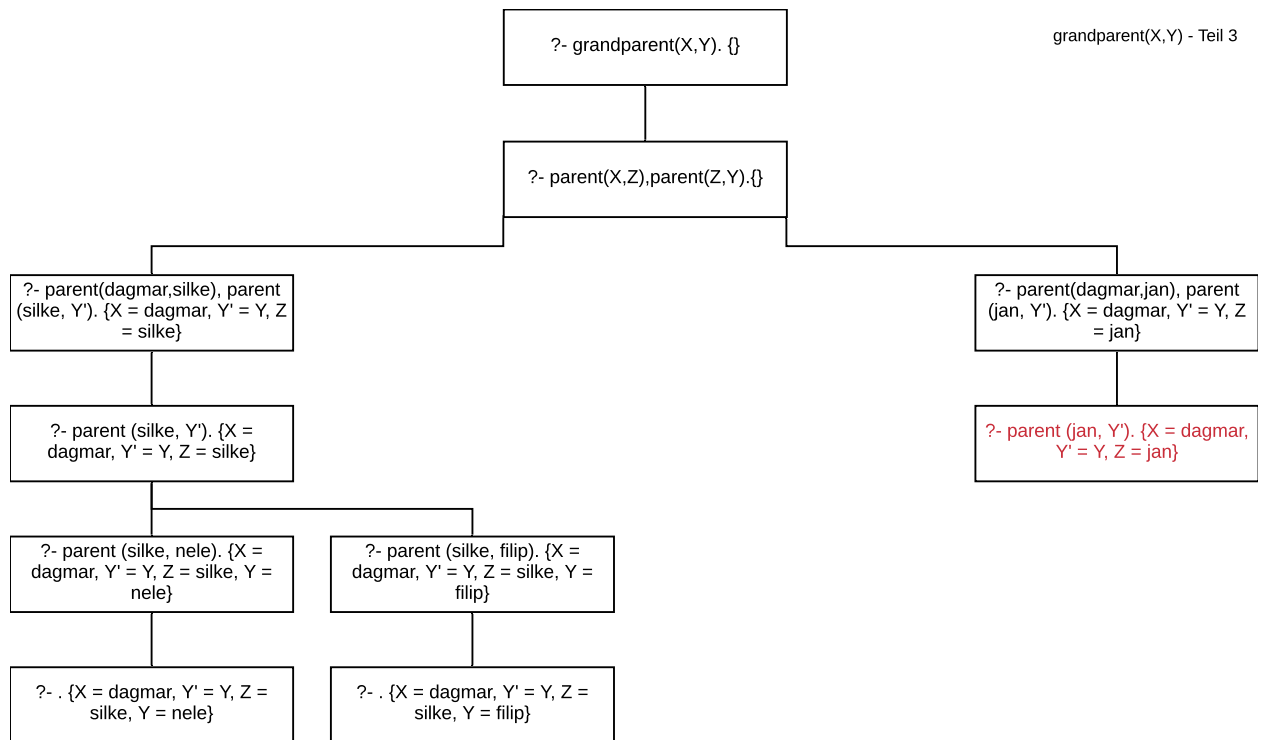


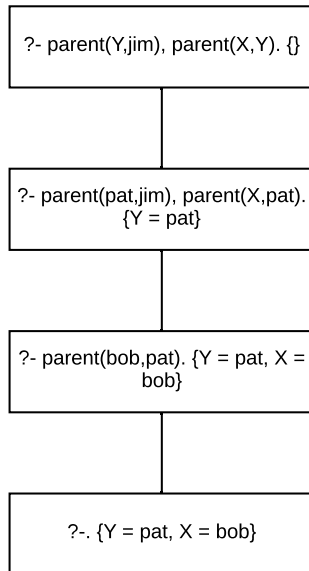
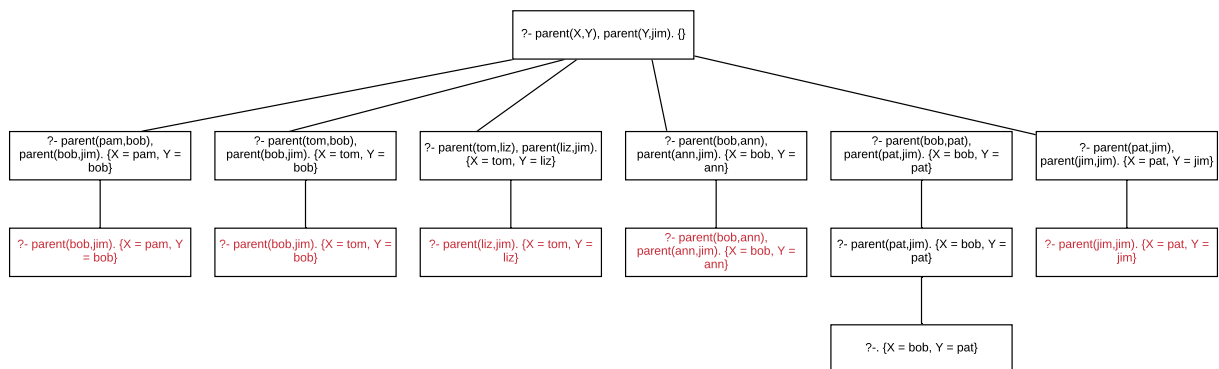
uncle(X,Y) - Teil 3









Aufgabe 2**2.A****2.B**Aufgabe 4**4.A**

```
[?- prod(X,Y,Z).
X = Z, Z = z ;
X = s(z),
Y = Z ;
X = s(s(z)),
Y = Z, Z = z ;
X = Z, Z = s(s(z)),
Y = s(z) ;
X = Y, Y = s(s(z)),
Z = s(s(s(s(z)))) .

[?- prod(s(s(z)),Y,Z).
Y = Z, Z = z ;
Y = s(z),
Z = s(s(z)) ;
Y = s(s(z)),
Z = s(s(s(s(z)))) ;
Y = s(s(s(z))),
Z = s(s(s(s(s(s(z)))))) ;
Y = s(s(s(s(z)))) ,
Z = s(s(s(s(s(s(s(s(z)))))))) .

[?- prod(X,Y,s(s(z))).
X = s(z),
Y = s(s(z)) ;
X = s(s(z)),
Y = s(z) ;
```

**Fehler:**

iii.) Prolog versucht alle möglichen Lösungen zu finden, in denen  $X \cdot Y = 2$  ist. Aufgrund fehlender Abbruchbedingung testet Prolog dieses Ergebnis auch mit  $X, Y \geq 2$ . Prolog versucht also alle Möglichkeiten durchzutesten, was zu einer Endlosschleife führt.