

# HW4

## Investigating Prolog

1. 

```
male(tom).
male(brian).
male(kevin).
male(zhane).
male(fred).
male(jake).
male(bob).
male(stephen).
male(paul).

parent(tom, stephen).
parent(stephen, jennifer).
parent(melissa,brian).
parent(mary,sarah).
parent(bob,jane).
parent(paul,kevin).
parent(tom,mary).
parent(jake,bob).
parent(zhane,melissa).
parent(stephen,paul).
parent(emily,bob).
parent(zhane,mary).
```

I put facts which related to tom and jennifer before other facts.
2. If we use the original order of facts, the rule will first try to match with fact: parent(tom,mary). However, parent(mary, jennifer) does not exist, thus the rule will try to match with another fact. In contrast, if we use the reordered facts, the rule will first match with the correct fact: parent(tom, stephen), then parent(stephen, jennifer). This could lead to a faster execution time.
- ```
[[trace] ?- grandfather(tom,jennifer).
[ Call: (8) grandfather(tom, jennifer) ? creep
[ Call: (9) male(tom) ? creep
[ Exit: (9) male(tom) ? creep
[ Call: (9) parent(tom, _6236) ? creep
[ Exit: (9) parent(tom, mary) ? creep
[ Call: (9) parent(mary, jennifer) ? creep
[ Fail: (9) parent(mary, jennifer) ? creep
[ Redo: (9) parent(tom, _6236) ? creep
[ Exit: (9) parent(tom, stephen) ? creep
[ Call: (9) parent(stephen, jennifer) ? creep
[ Exit: (9) parent(stephen, jennifer) ? creep
[ Exit: (8) grandfather(tom, jennifer) ? creep
[true .

[[trace] ?- grandfather(tom,jennifer).
[ Call: (8) grandfather(tom, jennifer) ? creep
[ Call: (9) male(tom) ? creep
[ Exit: (9) male(tom) ? creep
[ Call: (9) parent(tom, _6236) ? creep
[ Exit: (9) parent(tom, stephen) ? creep
[ Call: (9) parent(stephen, jennifer) ? creep
[ Exit: (9) parent(stephen, jennifer) ? creep
[ Exit: (8) grandfather(tom, jennifer) ? creep
[true .
```
3. No, I can't. Because I will need a rule: female(X) :- + male(X). However, a true + expression generally indicates inability to prove---not falsehood. Moreover, the universe of facts of facts is not complete. Therefore, I can not let the semantics of the not "male" to mean "female".
4. See hw4--mc7805--investigate.pl

## Unification

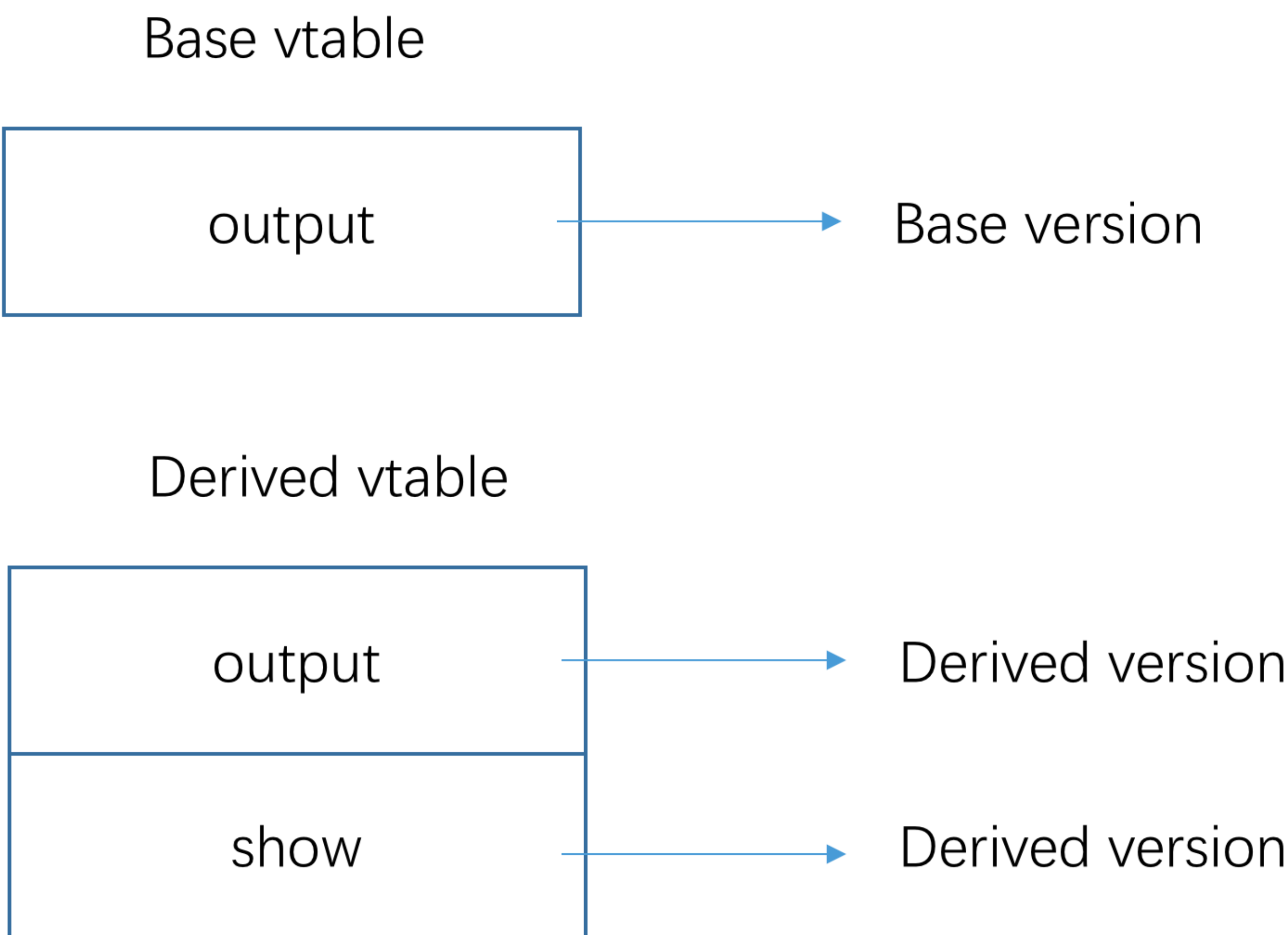
1. d(15) & c(X)                      d and c are different functors.
2. 42 & 23                              42 and 23 are different values.
3. a(X, b(3, 1, Y)) & a(4, Y)              Infinite recursion of Y.
4. a(X, c(2, B, D)) & a(4, c(A, 7, C))                      X=4; A=2; B=7; D=C
5. a(X, c(2, A, X)) & a(4, c(A, 7, C))                      It does not unify because A equals 2 and 7 at the same time.
6. e(c(2, D)) & e(c(8, D))                      The arguments of c are different.
7. X & e(f(6, 2), g(8, 1))                      X = e(f(6, 2), g(8, 1))
8. b(X, g(8, X)) & b(f(6, 2), g(8, f(6, 2)))                      X = f(6, 2)
9. a(1, b(X, Y)) & a(Y, b(2, c(6, Z), 10))                      The arity of b are different.
10. d(c(1, 2, 1)) & d( c(X, Y, X))                      X = 1; Y=2

## Virtual Functions

1. 

```
output base class
show base class
output derived class
show derived class
output derived class
show base class
output base class
show base class
```
2. 

```
b.output();
d.output();
d.show();
bp->output();
bp2->output();
```
3.



## Prototype OOLs

1. x
2. x y
3. x y z
4. x
5. 20
6. 20
7. 20
8. 10
9. \
10. 5
11. 5
12. 30