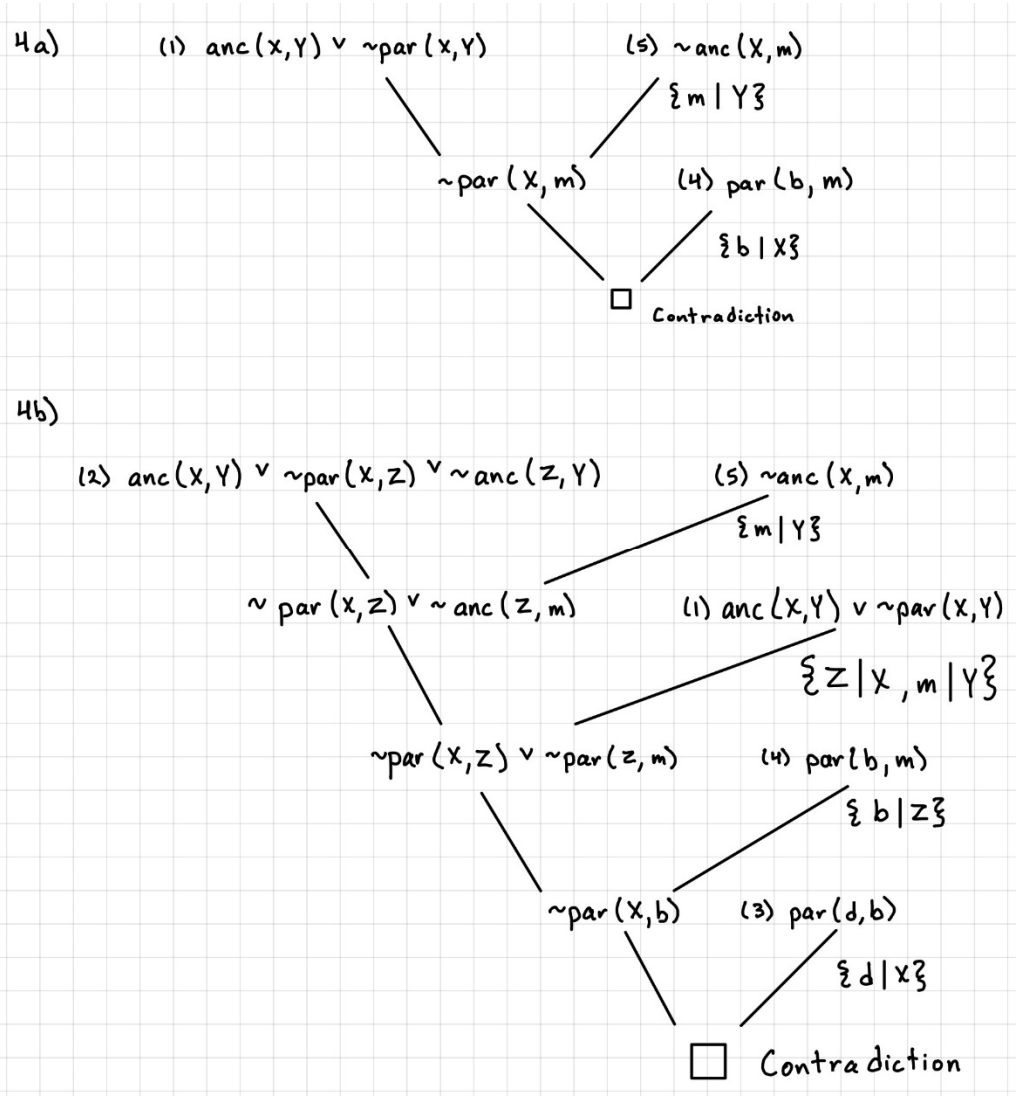
1. Complete the following sentences:
    a. Logic programming systems are also called deductive databases.
    b. The process of pattern matching to make statements identical is called unification.
2. Give a concise answer to each question below:
    a. What are the differences between procedural programming and logic programming?

	Procedural Programming	Logic Programming
Architecture	Von Neumann Machine (sequential steps)	Abstract model (dealing with objects and their relationships)
Syntax	Sequence of statements	Logic formulas (Horn Clauses)
Computation	Sequential statements execution	Deduction of the clauses
Control	Logic and control are mixed together	Logic and control can be separated

    b. What are the deficiencies of Prolog?
        i. Resolution order control
            1. Ordering of pattern matching during resolution
            2. Cut operator
        ii. Closed world assumption
            1. It has only the knowledge of its database
            2. A true/fail system rather than a true/false
        iii. The negation Problem
            1. Prolog not operator is not equivalent to logical NOT operator
    c. What are the motivations for Logic programming?
        i. Logic is used to represent program.
        ii. Deductions are used as computation.
        iii. A higher level language does more automatically – we can concentrate more on what is to be done and less on how to do it.
        iv. Ideal: Algorithm = logic (what) + Control (how) – only specify logic and let system take care of control
3. Use the set notation to describe resolution as a refutation system.
    a.

Given a set of clauses S & and goal G,
*negate the goal G
*$\{S\} \cup \{\neg G\}$
*existence of contradiction => derivation of empty clause
Based on $\{S\} \cup \{\neg G\}$ is inconsistent if $\{S\} \cup \{G\}$ is consistent

4. **Give deduction trees of resolution** (a) using (1) and (5); (b) using (2) and (5) -- Sample solution is in Canvas as "UnificationEx2" for the following set of clauses. Show each level of unification with instantiation (for example {m|Y}).
    (1) anc (X, Y) ᵛ ~par (X, Y)
    (2) anc(X, Y) ᵛ ~par (X, Z) ᵛ ~anc (Z, Y)
    (3) par (d, b)
    (4) par (b, m)
    (5) ~anc (X, m)



5. **Conjunctions and Backtracking**. Using the example of "Who teaches what" (see LogicProglecture page 20 in Canvas),
   (a) try to trace through search process for Query 2;
   (b) try to trace through Query 1, but with sub-goals reversed.

Who teaches what.

```
f1: teaches(john, database, s1).
f2: teaches(john, ai, s1).
f3: teaches(mary, os, s1).
f4: teaches(mary, ai, s2).
f5: teaches(paul, compiler, s1).
```

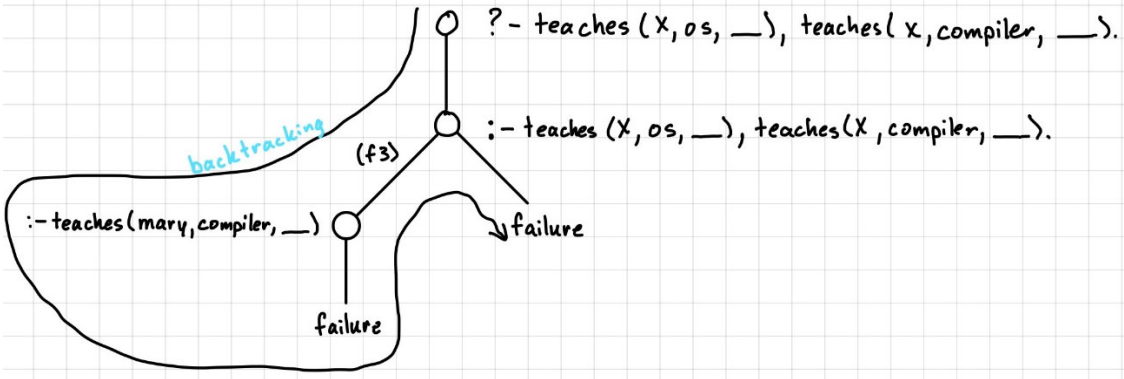
Query 1: What's the course taught by John and Mary?

```
?- teaches(john, X, _), teaches(mary, X, _).
```

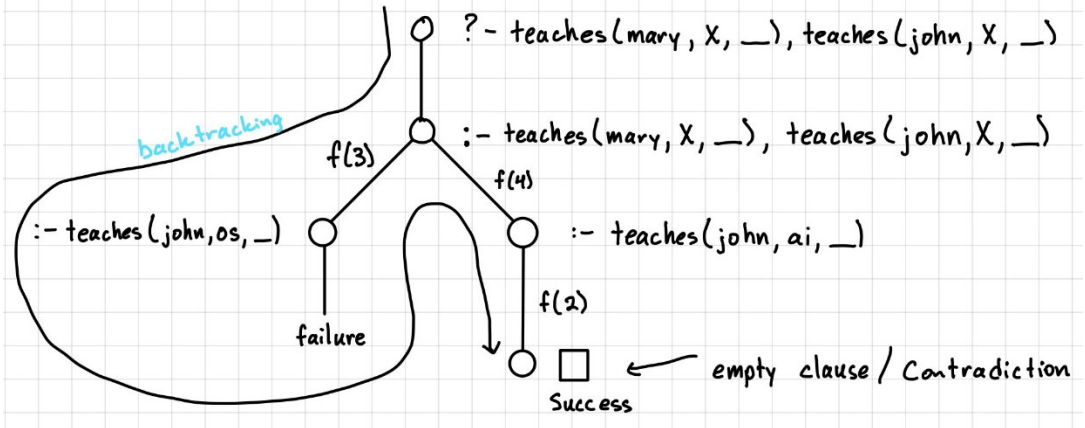
Query 2: Who teaches both os and compiler?

```
?- teaches(X, os, _), teaches(X, compiler, _).
```

5a) Try to trace through search process for Query 2.



5b) Try to trace through Query 1, but with sub-goals reversed.



6. **Exercise problem contribution.** Using the Example "Every scientist is logician" (see Canvas ExecofProlog example) as a guide, to create a problem with following 4 parts and then give solution to your own problem. **Post** your **problem** ((a) - (c)) and **solution** ((d) - (e)) **at your website** to share with your classmates. (Note: You may scan/take a picture of hand drawn deduction tree or draw it with a tool digitally)

   (a) Write a PROLOG representation of the following facts: (your at least 5 facts in English);

   (b) Write a PROLOG representation of the following rule: (your at least 2 rules in English);

   (c) Write two PROLOG goals statements to search for answers: (also give 2 W questions in English), and at least one of your goal statements should be a conjunction of two subgoals;

   (d) Run each given query in (c) using Prolog and then **post the interactive sessions as part of your solution at your website**;

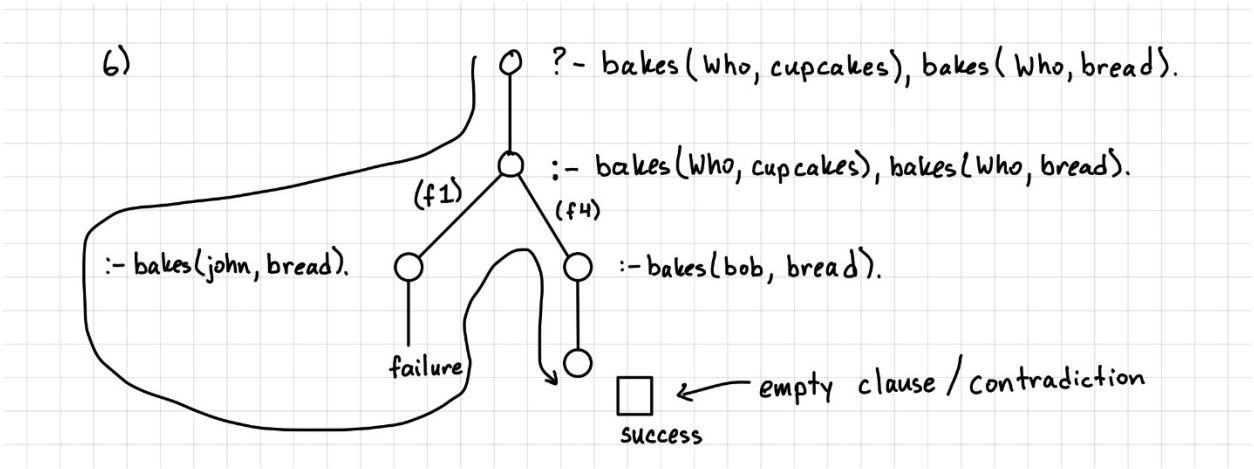
   (e) Show deduction tree that deducing the answer for one of the W (Who, What, Which, What) questions above according to Prolog search strategy (**a picture to post).**

Facts	**English meanings of Facts, Rules & Goals**
bakes(john, cupcakes).	// John bakes cupcakes
bakes(bob, bread).	// Bob bakes bread
bakes(sam, cookies).	// Sam bakes cookies
bakes(bob, cupcakes).	// Bob bakes cupcakes
chef(john).	// John is a chef
chef(sam).	// Sam is a chef
baker(bob).	// Bob is a baker
Rules	
taught(bob,Person) :- chef(Person).	// Bob taught a Person if they are a chef.
experiments(Who,What):- bakes(Who,What).	// If someone bakes something, he/she experiments with it.
Queries / Goals & answers	
?- bakes(Who,cupcakes),bakes(Who,bread). Who = bob.	// Who bakes both cupcakes and bread? Answer: bob
?- taught(Who,sam). Who = bob	// Who taught Sam to bake? Answer: bob
?- taught(bob,sam). true.	// Did Bob teach Sam to bake? Answer: true

?- experiments(bob,What). What = bread.	// Bob experiments with what? Answer: bread
?- experiments(Who,bread). Who = bob.	// Who experiments with bread? Answer: bob
?- experiments(Who,cupcakes). Who = john.	// Who experiments with cupcakes? Answer: john

```
/* Facts */
bakes(john, cupcakes).
bakes(bob, bread).
bakes(sam, cookies).
bakes(bob, cupcakes).
chef(john).
chef(sam).
baker(bob).

/* Rules */
taught(bob,Person) :- chef(Person).
experiments(Who,What) :- bakes(Who,What).
```

```
?- bakes(Who,cupcakes),bakes(Who,bread).
Who = bob.

?- taught(Who,sam).
Who = bob.

?- taught(bob,sam).
true.

?- experiments(bob,What).
What = bread ;

?- experiments(Who,bread).
Who = bob.

?- experiments(Who,cupcakes).
Who = john ;
```


6)
?- bakes(Who, cupcakes), bakes(Who, bread).
:- bakes(Who, cupcakes), bakes(Who, bread).
(f1)
(f4)
:- bakes(john, bread).
:- bakes(bob, bread).
failure
empty clause / contradiction
success