1. (8 %) 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. (15%) G**ive a concise answer to each questions below:**

(a) What are the differences between procedural programming and logic programming?

In procedural programming, logic and control are mixed together whereas in logic programming logic and control are separated (control being for the most part hidden from the programmer). Logic programs focus on what a problem is, not on how to solve it (you leave that to the computer). Procedural programs are sequential (influenced by the Von Neumann Model) while logic programs don't really have a sequence, but rather a set of rules and relationships from which something can be deduced. The facts and relationships in logic programming are expressed with Horn Clauses. Logic programming lacks control on the programmer's end: clauses can be in any order (they are an unordered set) and control happens inside the system.

(b) What are the deficiencies of Prolog?

Prolog has only the knowledge of the database supplied to it, which is known as "closed world". It bases results on true or fail, instead of true or false, where fail means that the answer might exists, but we couldn't find out. The 'not' operator in prolog is not quite the same as a logical 'not' (e.g. given parent(michelle, chris), the outcome of not(mother(michelle,chris)) could be true even though michelle couldn't be Chris's father. The computer might be lacking the key fact that female(michelle)).

(c) What are the motivations for Logic programming?

We can focus on what is to be done (what's important, i.e. the problem) and not worry about how it is to be done/solved. We only specify logic and let the system take care of control.

3. (7%) Use the set notation to describe resolution as a refutation system.

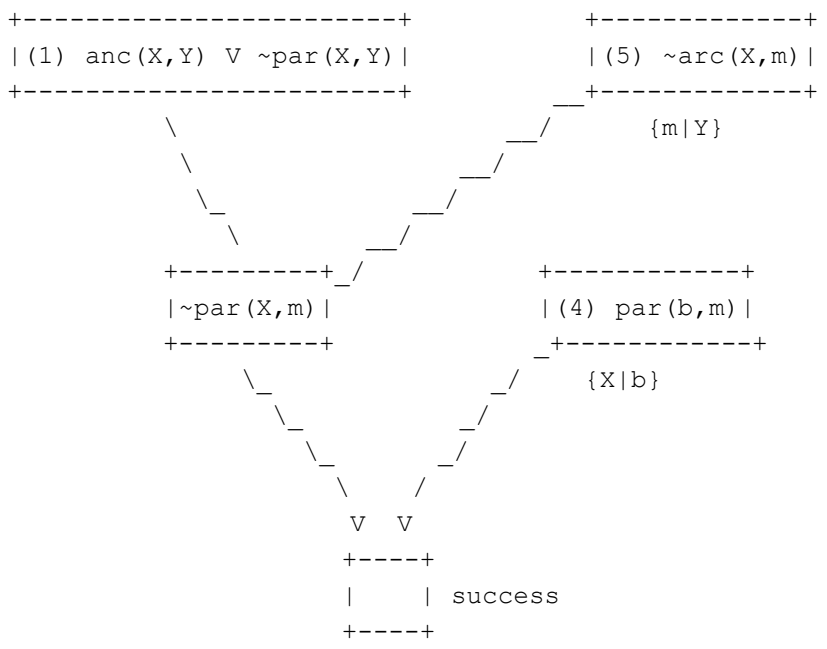
Refutation is when you prove something is true by negating it then showing that the negation is inconsistent with the premises.

Given a set of clauses S and a goal G:

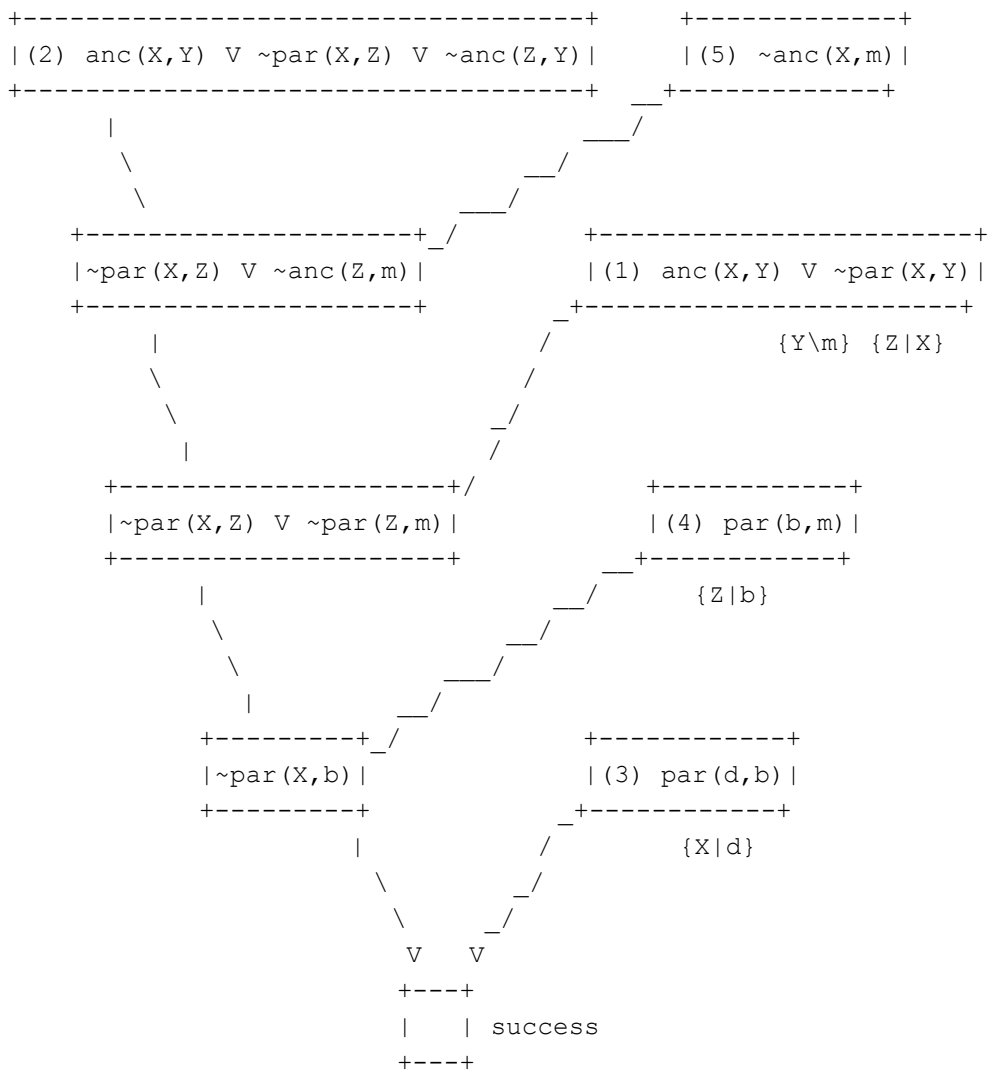
- negate the goal G
- join {S} U {¬G}
- The existence of contradiction => derivation of empty clause (the original goal is true)

{S} U {¬G} is inconsistent if {S} U {G} is consistent.

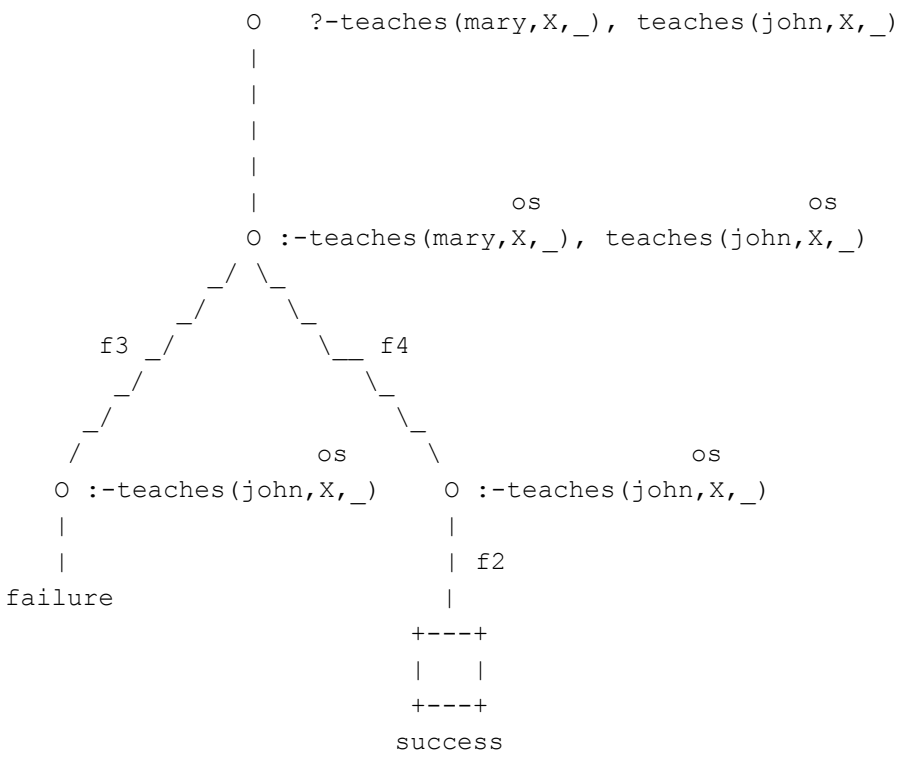
4a.


(1) anc(X,Y) V ~par(X,Y)
(5) ~arc(X,m)
{m|Y}
~par(X,m)
(4) par(b,m)
{X|b}
success

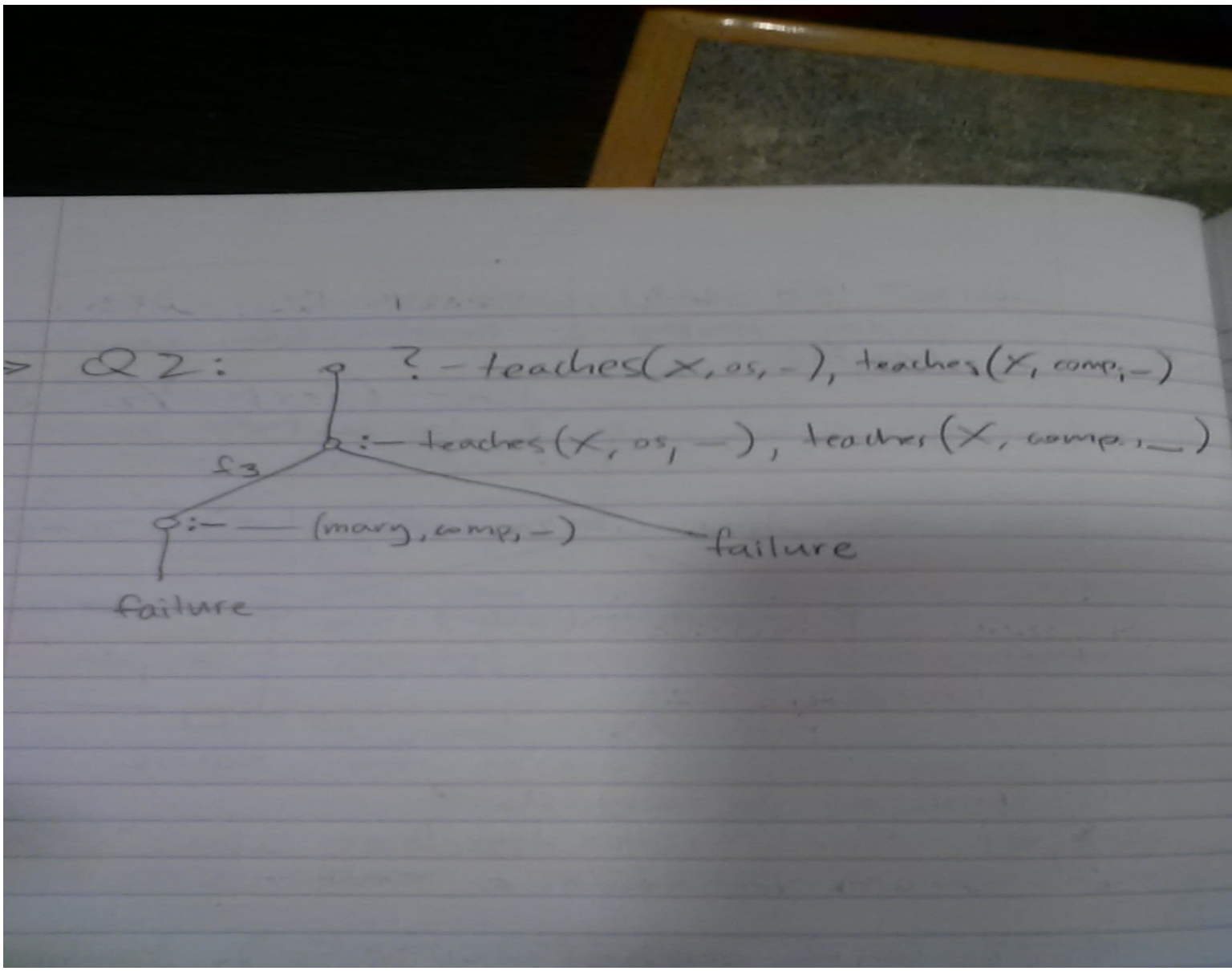
4b.


(2) anc(X,Y) V ~par(X,Z) V ~anc(Z,Y)
(5) ~anc(X,m)
~par(X,Z) V ~anc(Z,m)
(1) anc(X,Y) V ~par(X,Y)
{Y\m} {Z|X}
~par(X,Z) V ~par(Z,m)
(4) par(b,m)
{Z|b}
~par(X,b)
(3) par(d,b)
{X|d}
success

5b.


?-teaches(mary,X,_), teaches(john,X,_)
os
os
:-teaches(mary,X,_), teaches(john,X,_)
f3
f4
os
os
:-teaches(john,X,_)
:-teaches(john,X,_)
f2
failure
success

5a.


Q2:
?- teaches(X, os, _), teaches(X, comp, _)
:- teaches(X, os, _), teaches(X, comp, _)
f3
:- (mary, comp, _)
failure
failure