Homework Assignment 3

CSC520
James Skripchuk

Question 1

Sentences:

Sally will write if and only if she wants to publish.
$t \leftrightarrow q$
$(\neg t \lor q) \land (\neg q \lor t)$

If it is Tuesday, Sally wants to rest.
$a \rightarrow s$
$\neg a \lor s$

If Sally is reading, she either wants to rest or publish.
$p \rightarrow (s \lor q)$
$\neg p \lor s \lor q$

If Sally is not reading, then it is not Tuesday.
$\neg p \rightarrow \neg a$
$p \lor \neg a$

Sally is reading.
p

Let us go with the following assignment:
p=T, q=T, s=F, a=F, t=T

	p	$p \lor \neg a$	$\neg p \lor s \lor q$	$\neg a \lor s$	$(\neg t \lor q) \land (\neg q \lor t)$
Substitution	T	T \/ ~F	~T \/ F \/ T	~F \/ F	(~T \/ T) /\ (~T \/ T)
	T	T \/ T	F \/ F \/ T	T \/ F	T /\ T
Result	T	T	T	T	T

As we see, all the statements are satisfied, meaning that this assignment is a satisfactory model.

Question 2

Using the following lexicon, represent the following sentences in FOL. Use resolution and unification to determine if you can derive a contradiction for the target sentence.

Michelle, Sheryl and Berne are the members of WiCS.

$WiCS(Michelle)$

$WiCS(Sheryl)$

$WiCS(Berne)$

Each member of WiCS is either a Physics major, CS major or both.

$\forall x\ WiCS(x)\ \rightarrow Phy(x) \lor Com(x)$

CNF:

$\forall a\ \neg WiCS(a) \lor Phy(a) \lor Com(a)$

None of the CS majors study Mechanics

$\forall x\ Com(x)\ \rightarrow \neg enroll(x,\ Mech)$

CNF:

$\forall b\ \neg Com(b)\ \lor \neg enroll(b,\ Mech)$

and all the Physics majors take Scientific computing.

$\forall x\ Phy(x)\ \rightarrow enroll(x,\ SciComp)$

CNF:

$\forall c\ \neg Phys(c)\ \lor enroll(c,\ SciComp)$

Michelle is enrolled in courses that Sheryl is not,

$\forall c\ \neg enroll(Sheryl, c)\ \rightarrow enroll(Michelle, c)$

CNF:

$\forall d\ enroll(Sheryl, d)\ \lor enroll(Michelle, d)$

And (Michelle) is not in the courses that Sheryl is enrolled in.

$\forall c\ enroll(Sheryl, c)\ \rightarrow \neg enroll(Michelle, c)$

CNF:

$\forall e\ \neg enroll(Sheryl, e) \rightarrow \neg enroll(Michelle, e)$

Sheryl is not enrolled in Mechanics and Scientific Computing.

$\neg enroll(Sheryl, Mech)$

$\neg enroll(Sheryl, SciComp)$

Target

There is a WiCS member, who is a CS major but not a Physics major

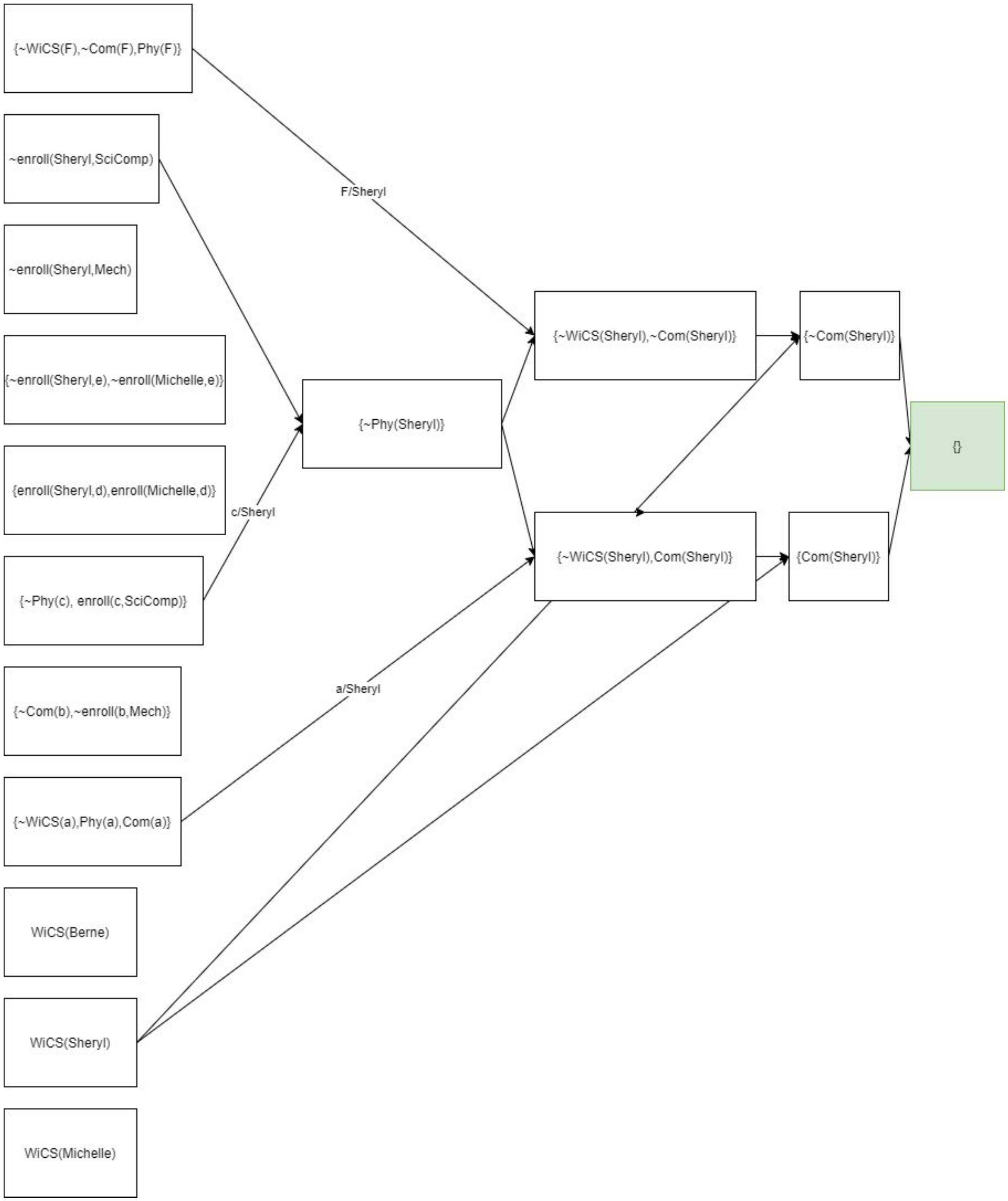
$\exists m\ WiCS(m) \wedge Com(m) \wedge \neg Phy(m)$

CNF:

$WiCS(M) \wedge Com(M) \wedge \neg Phy(m)$

Negated Conclusion:

$\neg(WiCS(M) \wedge Com(M) \wedge \neg Phy(m))$ = $\neg WiCS(M) \vee \neg Com(M) \wedge Phy(m)$



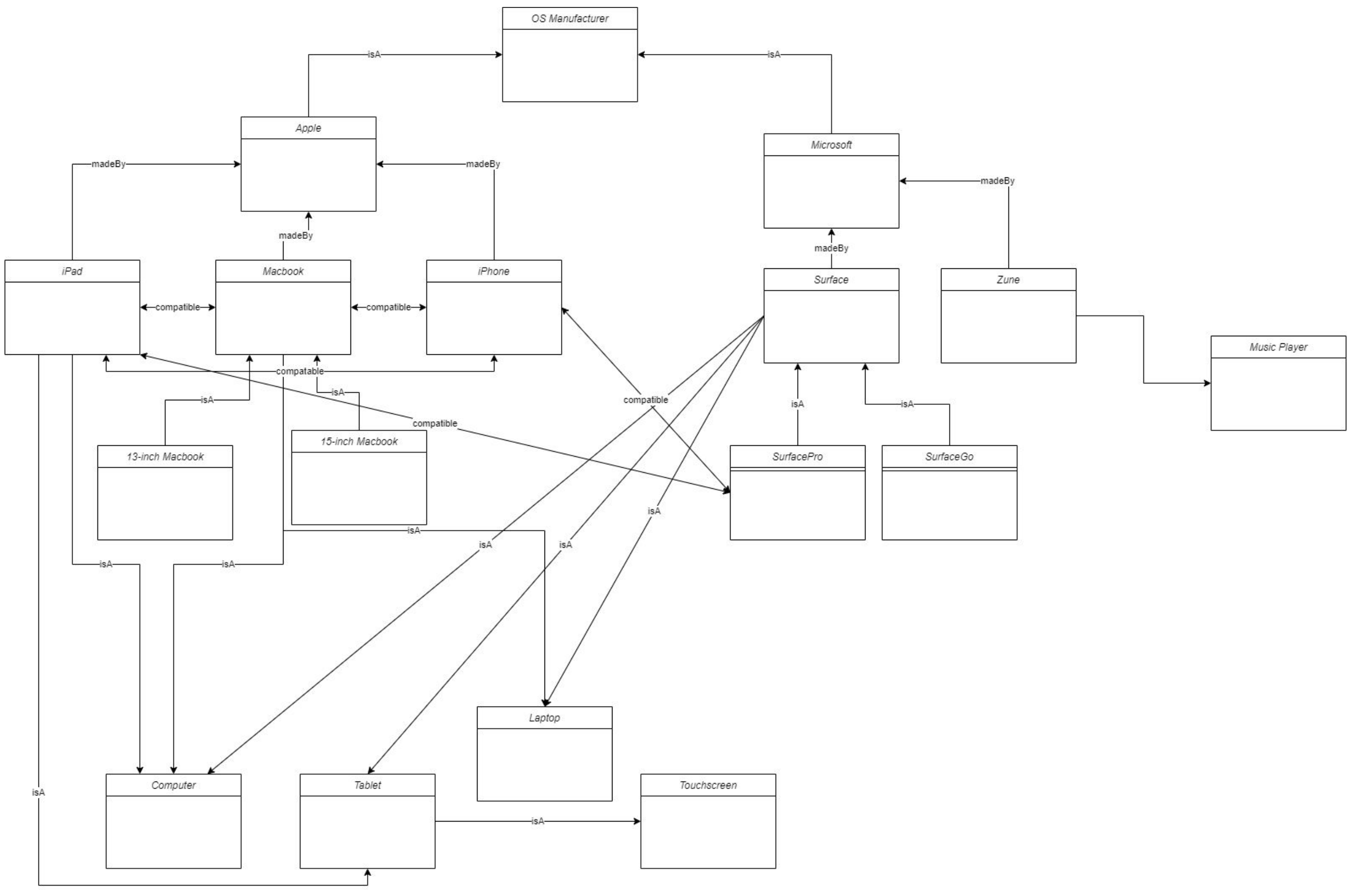
Here is the resolution tree for the above statement. All physics majors must take scientific computing. Since Sheryl is not enrolled in Scientific Computing, that means that she is not a physics major. Sheryl is also in WiCS. Since everyone in WiCS is either a Physics major or a CS major (or both), and using the fact that Sheryl is not a Physics major, we can deduce that Sheryl must be a CS major and a CS major alone. (Which satisfies our target)

Question 3

I've included the image separately in my .zip submission if you can't zoom in enough.

Here i'm assuming the closed world assumption (e.g. the lack of something such as touchscreens means that the relationship is false).

The relationships and their properties are in q3.pl.



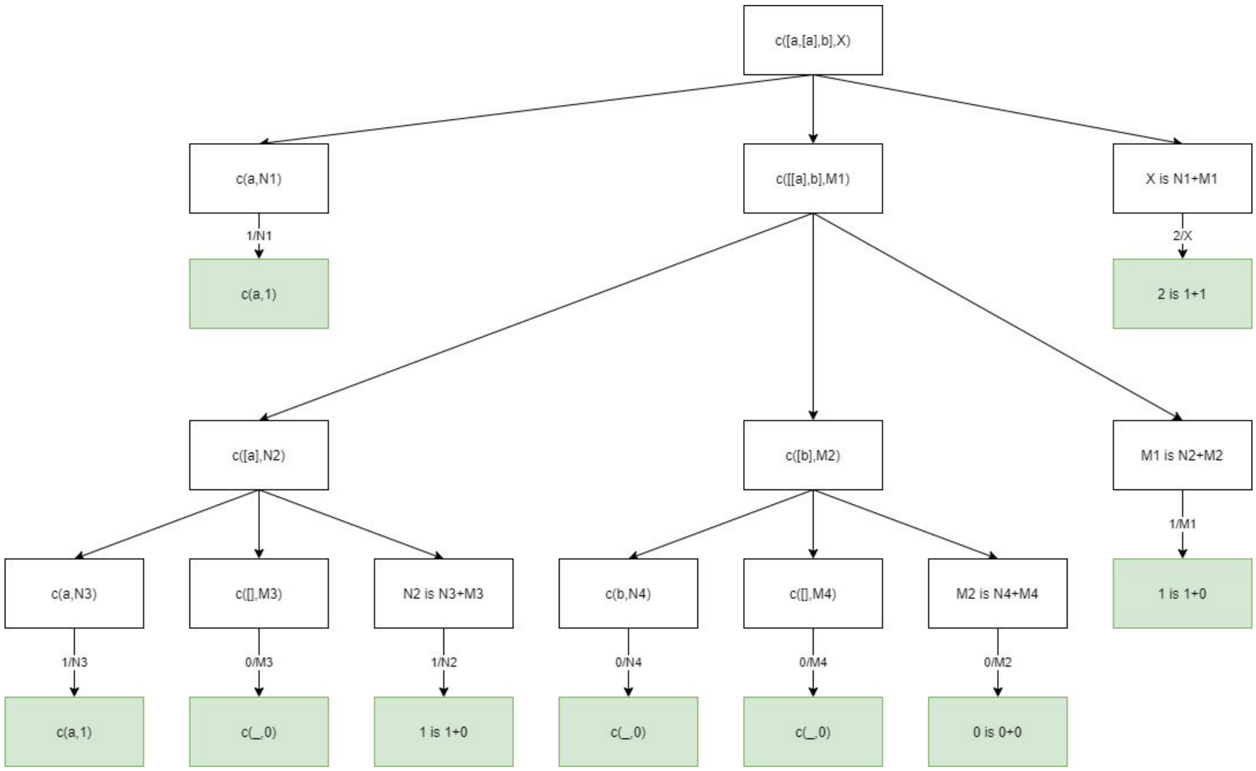
Question 4

What are the answers to this query? [c ([a , [a] , b] ,X).]

Since there are two "a's", the answer is **2**

List step by step how the answer is obtained by the Prolog interpreter. In each step, state the subgoal list, the line number of the clause matched, and the substitutions. See tree.

I find the tree format much more digestible, so here's an additional visualization I made of the matching process.



And here's the tabular method.
I've bolded the clauses that are matched.

Subgoal List	**MC**	**Substitutions**
c([a, [a], b], X)	1	{a/H, [[a],b]/T}
c(a, N 1), c([[a], b], M 1), X is N 1 + M 1	2	{1/N1}
c([a], N2), c([b],M2), M1 is N2+M2, X is 1 + M 1	1	
c(a,N3), c([], M3), N2 is N3+M3, c([b],M2), M1 is N2+M2, X is 1 + M 1	2	{1/N3}
c([], M3), N2 is 1+M3, c([b],M2), M1 is N2+M2, X is 1 + M 1	3	{0/M3}
N2 is 1+0, c([b],M2), M1 is N2+M2, X is 1 + M 1		{1/N2}
c([b],M2), M1 is 1+M2, X is 1 + M 1	1	
c(b,N4), c([],M4), M2 is N4+M4, M1 is 1+M2, X is 1 +	3	{0/N4}

M 1		
c([],M4), M2 is 0+M4, M1 is 1+M2, X is 1 + M 1	3	{0/M4}
M2 is 0+0, M1 is 1+M2, X is 1 + M 1		{0/M2}
M1 is 1+0, X is 1 + M 1		{1/M1}
X is 1 + 1		{2/X}

If we remove the cut term from the second line yielding a statement with no subclauses, Do the answer(s) change? If so, why?

When we remove the cut term from the second clause, the answer does not change, even though the program code changes slightly. For example, if we were to try and match c(a,1), it would first match with clause 2; but then prolog would redo and match with c(_,0), since _ matches with any term. However, due to the cut in the first clause, this never becomes an issue in our current problem.

What will happen if we subsequently remove the cut term of the first line? Why?

Now, if we now remove all cuts in the program, our solution changes dramatically. Instead of a single result of X=2, we get:

```
X = 2 ;
X = 2 ;
X = 1 ;
X = 1 ;
X = 1 ;
X = 1 ;
X = 1 ;
X = 1 ;
X = 1 ;
X = 0 ;
X = 0 ;
X = 0 ;
X = 0 ;
X = 0 ;
X = 0.;
```

Looking into the trace, we discover that the first solution proceeds similarly to the ones without the cuts. However, after the first solution of X=2 is found, prolog backtracks and tries to find all possible solutions that match the clauses. For example, similar to the previous question, one of matches c(a,1), but since it tries to go back and find *all solutions*, it backtracks and matches with c(_,0). Since _ matches with anything, we get many possible combinations of 1+1, 0+0, and

1+0, thus why there are so many repeated solutions and why they range from 0-2. The usage of the cut statements in the original formulation avoided this issue.

Question 5 & 6

See q5.pl and q6.pl