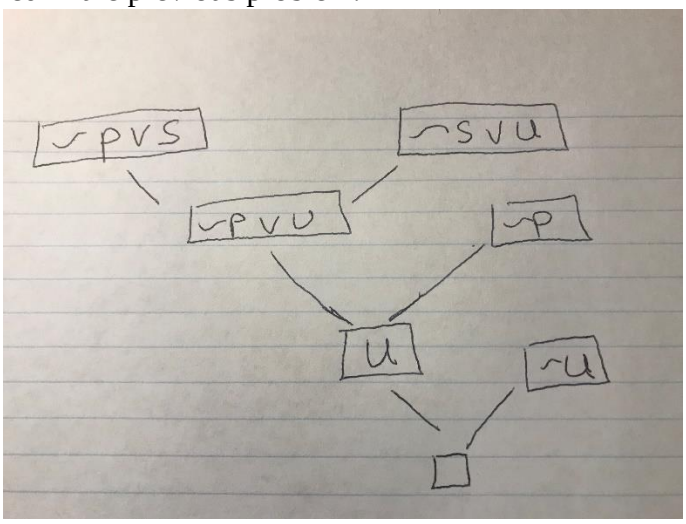
Giovanni Anastasio

Propositional Calculus Problem Set 2

1. State the resolution principle.
    a. The resolution principle states that " for any two clauses C1 and C2, if there is a literal Li in Cl and a complementary literal L2 in C2, then delete L1 from C1 and delete L2 from C2 and construct the disjunction of the remaining clause. The disjunction of the remaining clauses is called the resolvent and is a logical consequence of C1 and C2.
2. Define what is meant by resolution deduction.
    a. Given a set S of clauses, a resolution deduction of clause C from set S is a finite sequence C1, C2, C3… Ck of clauses such that each Ci is either a clause of S or a resolvent of two preceding clauses, and C = Ck.
3. Show by means of resolution, line by line, that the formula U is a logical consequence of the three formulas (P → S) and (S → U) and P. (Please don't forget that you must first convert to casual form.)
    a. Re-write as a set of clauses
        i. S = { ~P ∨ S, ~S ∨ U, P }
    b. Add the negated goal
        i. S = { ~P ∨ S, ~S ∨ U, P , ~U }
    c. The refutation

i.	~P ∨ S	(Element of S)
ii.	~S ∨ U	(Element of S)
iii.	P	(Element of S)
iv.	~U	(Element of S)
v.	~P ∨ U	(Resolution of ii and i)
vi.	U	(Resolution of iii and v)
vii.	[]	(Resolution of iv and vi)

4. Draw the resolution tree that corresponding to the resolution deduction that you performed in the previous problem.
    a. 

5. Show by means of the inconsistency truth table approach that the formula U is a logical consequence of the three formula (P → S) and (S → U) and P.

P	**S**	**U**	**(P → S)**	**(S → U)**	**(P → S) ∧ (S → U)**	**(P → S) ∧ (S → U) ∧ P**
T	T	T	T	T		
T	T	F	T	F		
T	F	T	F	T		
T	F	F	F	T		
F	T	T	T	T		
F	T	F	T	F		
F	F	T	T	T		
F	F	F	T	T		

6. Define Horn clause.
    a. A Horn clause is a clause which at most has one unnegated literal.
7. Can the formula (P ∧ Q ∧ R) → S be converted to a Horn clause?
    a. Yes, by applying the switcheroo rule.
8. Write down a clause which is not Horn clause.
    a. (~R ∧ ~Q)
9. Write down a Horn clause involving P, Q, R, S and T.
    a. (P ∧ Q ∧ R) ∨ (S ∧ T)
10. Argue that the following prolog statement is a Horn clause: a :- b, c, d.
    a. The prolog statement above translates to: (b ∧ c ∧ d) → a. The comma's in the statement stand for conjunctions and the :- refers to an implication. By using a couple rules that logic statement can be transformed into a horn clause.
11. Prolog is essentially a Horn clause problem solver. (True or False)
    a. True
12. Prolog performs a computation by (1) converting its rules to Horn clauses, (2) negating a given goal, and (3) endeavoring to derive the empty clause by means of resolution. If variables are involved, they must be instantiated in order to obtain complementary literals. (True or False)
    a. True
13. Consider the following prolog program:

    p :- q, t.

    p :- r, s.

    p :- r.

    r.

    t.

    s.

a. Convert to Horn clause form. (simply write down each statement as a Horn clause.)
    i. ~Q ∨ ~T ∨ P
    ii. ~R ∨ ~S ∨ P
    iii. ~R ∨ P
    iv. R
    v. T
    vi. S
b. Draw a resolution deduction tree which derives the empty clause, assuming that P is the goal.
c. Draw a different resolution deduction tree which derives the empty clause.

14. Consider the following prolog program:

% Animal KB

% Think of these rules as what is known about animals.

Id (polarbear) :- description(large , white).

Id (dove) :- description(small , white).

Id (cow) :- description(large , brown).

Id (chipmunk) :- description(small , brown).

Id (cardinal) :- description(small , red).

description(Size, Color) :- size(Size), color(Color).

% Think of these facts as what we see in the wild.

a. Draw a resolution deduction tree, instantiating variables as needed, to respond to the prolog query, add it to the set of clauses (the Prolog program in clausal form), and refute (instantiating variables appropriately)!

15. With the help of Google, find out an interesting big picture Something about Prolog, and simply write it down.
    a. Something interesting that I found out about Prolog as a whole is it is one of the first logic programming languages, developed in 1972. It remains to be the most popularly used logical programming language to work on natural language processing. Prolog also has one data type called a term and a term can be either atoms, numbers, variables, or compound terms.