Homework 1

Group members: *Tushar Yadav, Dhruti Dilipbhai Patel, Pratishtha Soni*

1. Provide a "group picture" of your group.

2. Determine whether the following statements can be represented as a Proposition. (Answer yes or no)
   a. Fullerton is in California. **Yes**
   b. $2 < 5$: **Yes**
   c. Every course has prerequisites: **Yes**
   d. A prerequisite of CPSC 583: **No**
   e. CPSC 481 is a prerequisite of CPSC 583: **Yes**
   f. All dogs can be pets: **Yes**

3. Determine whether the following sentences are well-formed formulae in Propositional Logic. (Answer yes or no)
   a. $P \rightarrow (Q \vee (R \rightarrow S))$ **Yes**
   b. $P \leftarrow Q$ **No**
   c. $\sim Q \wedge Q$ **Yes**
   d. $\sim\sim\sim Q$ **Yes**
   e. $\forall x\, P(x) \rightarrow Q(x)$ **No**

4. Convert the following propositional logic sentences to CNF (Give the answers as a numbered list of clauses)
   a. $P \rightarrow (Q \rightarrow R)$
      i. ***P → (Q → R)***
      ii. ***~P v (Q → R)***
      iii. ***~P v (~Q v R)***
      iv. ***~P v ~Q v R***

   b. $\sim(P \rightarrow R)$
      i. ***~ (P → R)***
      ii. ***~ (~P v R)***
      iii. ***P ∧ ~R***
      iv. ***P, ~R***

   c. $P \rightarrow (R \wedge S)$
      i. ***P → (R ∧ S)***
      ii. ***~P v (R ∧ S)***
      iii. ***(~P v R) ∧ (~P v S)***
      iv. ***(~P v R), (~P v S)***

   d. $\sim(Q \vee R)$
      i. ***~(Q v R)***
      ii. ***~Q ∧ ~R***
      iii. ***~Q, ~R***

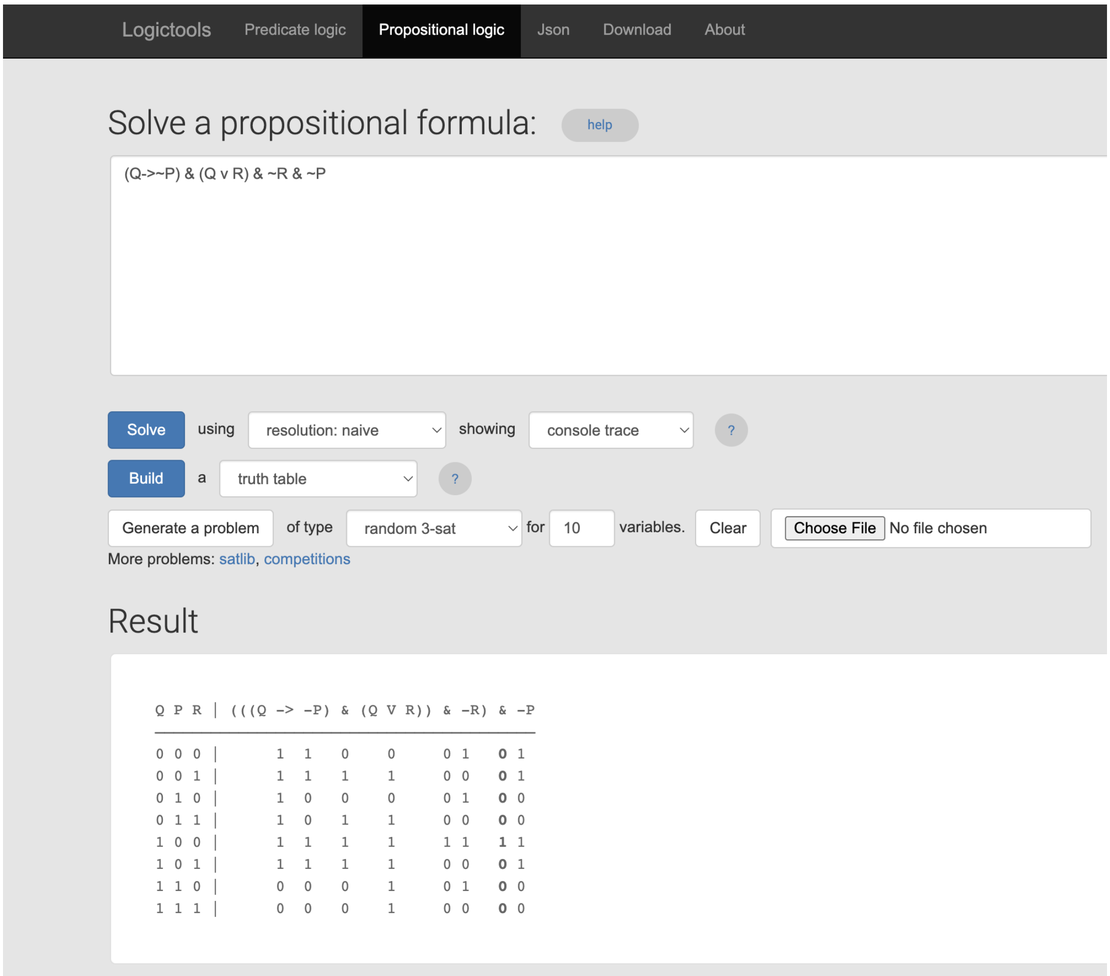
   e. $P \rightarrow (Q \vee (R \rightarrow S))$
      i. ***P → (Q ∨ (R → S))***
      ii. ***~P ∨ (Q ∨ (~R ∨ S))***
      iii. ***~P ∨ (Q ∨ ~R) ∨ (Q ∨ S)***
      iv. ***~P ∨ Q v ~R ∨ S***

5. Attempt to resolve the given pairs of sentences in CNF. Either write the resulting sentence, or state that they cannot be resolved.
   a. $\sim P \vee Q \vee R$, $P \vee Q \vee R$ **=** ***Q ∨ R***
   b. $\sim P \vee Q \vee R$, $P \vee Q \vee S$ **=** ***Q ∨ R ∨ S***
   c. $\sim P \vee Q \vee R$, $\sim R$ **=** ***~P ∨ Q***

6. Determine whether the following Propositional Logic conclusions are valid or invalid.
   a. Premises: Q→~P, Q∨R, ~R; Conclusion: ~P
      i. Prove using **a truth table**

P	~P	Q	R	Q→~P	Q∨R	~R	~P
T	F	T	T	F	T	F	F
T	F	T	F	F	T	T	F
T	F	F	T	T	T	F	F
T	F	F	F	T	F	T	F
F	T	T	T	T	T	F	T
F	T	T	F	T	T	T	T
F	T	F	T	T	T	F	T
F	T	F	F	T	F	T	T

      ii. Verify the answer using any of the solvers at

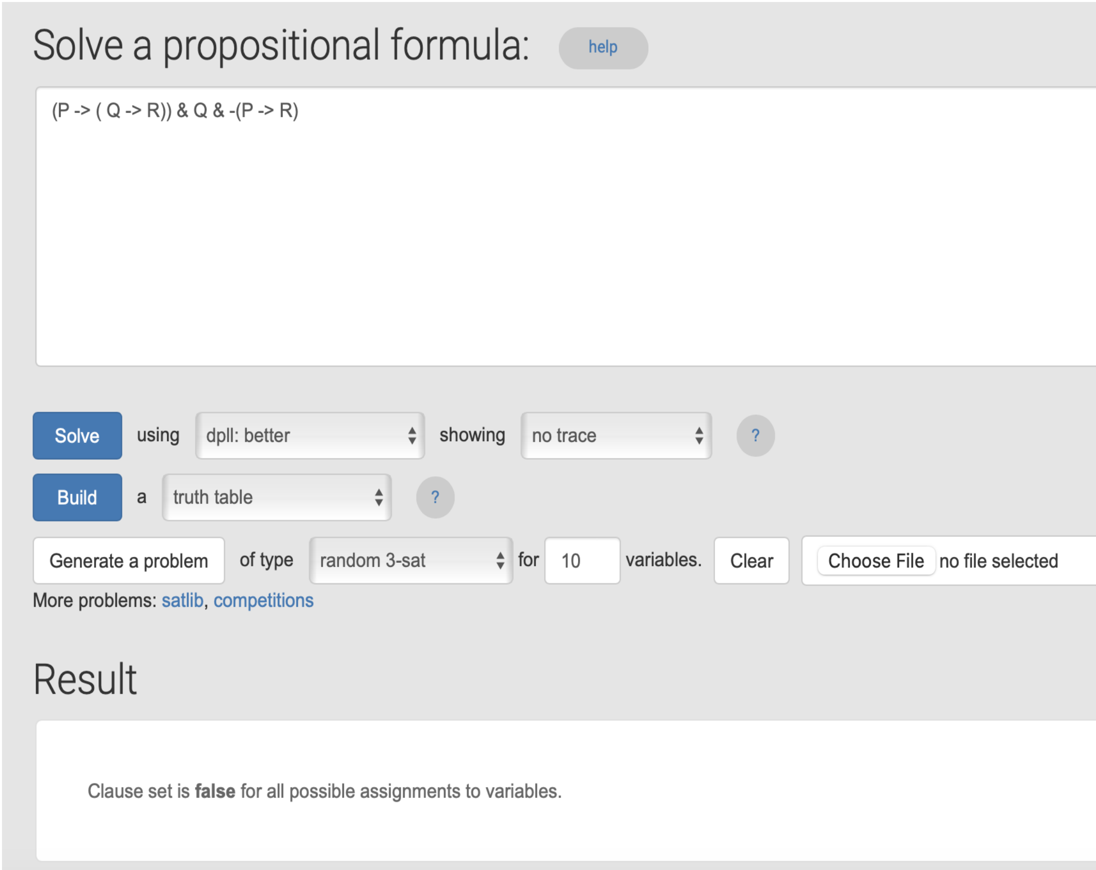


b. Premises: P →(Q→R), Q; Conclusion: P → R

i. Prove using resolution refutation

1. ~P v ~Q v R
2. Q
3. P
4. ~R
5. ~P v R [Res. 1, 2]
6. R [Res. 3, 5]
7. FALSE: [Contradiction] - Conclusion is valid

ii. Verify the answer using any of the solvers at

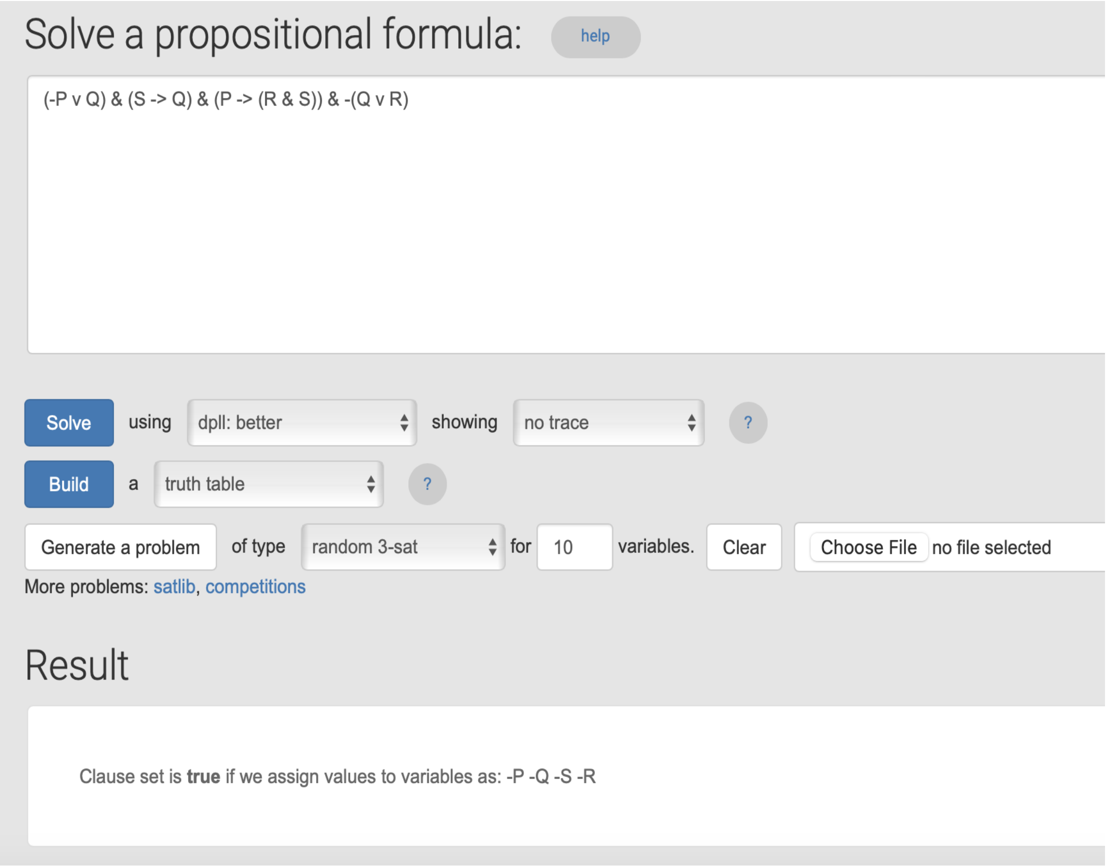


c. Premises: ~P ∨ Q, P→ (R ∧ S), S→Q; Conclusion: Q ∨ R

i. Prove using resolution refutation

1. ~P v Q
2. ~P v R
3. ~P v S
4. ~S v Q
5. ~ Q
6. ~ R
7. ~P [Res. 1,5]
8. ~S [Res. 4,5]
9. [Can't apply more] - Conclusion can't be proved

ii. Verify the answer using any of the solvers at



7. The "length" of a clause in CNF is the number of literals in it. For example, (~P ⋁ Q) has length 2, and (~P ⋁ Q ⋁ ~R ⋁ S) has length 4. The following questions are about the length of the result clause after resolving a pair of clauses.

    a. If the given pair of clauses each have length 2, what will be the length of the result clause?

    Ans: 2; If you have two clauses of length 2 each and you perform a resolution step on them, the resulting clause will typically have a length of 2.

    Example:

    i. (A ⋁ B)
    ii. (~A ⋁ C)
    Iii. B ⋁ C – resolving (i & ii)

    b. If the given pair of clauses each have length 3, what will be the length of the result clause?

    Ans: 3 or 4;

    Example:

    i. (A ⋁ B ⋁ C)
    ii. (~A ⋁ D ⋁ E)
    iii. (B ⋁ C ⋁ D ⋁ E) – resolving i & ii

c. If the given pair of clauses each have length 4, what will be the length of the result clause?

Ans: Maximum 8 & Greater than 4;

Example:

i. (A ∨ B ∨ C ∨ D)
ii. (~A ∨ E ∨ F ∨ G)
iii. (B ∨ C ∨ D ∨ E ∨ F ∨ G) – resolving i & ii

or

i. (A ∨ B ∨ C ∨ D)
ii. (E ∨ F ∨ G ∨ H)
iii. (A ∨ B ∨ C ∨ D ∨ E ∨ F ∨ G ∨ H) – resolving i & ii

d. In general, will the length of the result clause be longer, shorter, or the same length as the given pair of clauses?

Ans:

Typically, when you perform a resolution step on a pair of clauses, the resulting clause's length will either be shorter or the same as the original pair of clauses. It won't be longer. Since, achieving shorter length is the most common outcome when resolving clauses.

8. A propositional **2-CNF** expression is a conjunction of clauses, each containing at most *2* literals, e.g., A ∨ B, ~A ∨ C, ~B ∨ D, ~C ∨ G, ~D ∨ G

a. Prove using resolution that the above sentence entails G.

ANS

1. ***A v B***
2. ***~ A v B***
3. ***~ B v D***
4. ***~ C v G***
5. ***~ D v G***
6. ***B [Res. 1, 2]***
7. ***D [Res. 3, 6]***
8. ***G [Res. 5, 7]***
9. ***Hence, proved that, the above sentence entails G.***

b. How many distinct 2-CNF clauses can be constructed from ***n*** propositions?

ANS

Total number of combinations that can be formed with all cases from 2CNF is n(2n-1).

Total: n(2n-1)

If we consider clauses with the same literals like A, B, ~A then we add 2n.

Total: n(2n-1) + 2n

And if we consider cases that are equivalent then we need to subtract (n-1) cases from total.

$$\textbf{\textit{Total: n(2n-1) + 2n - (n-1)}} = 2n^2 + 1$$

c. Using your answer to (b), prove that propositional resolution for a **2-CNF** expression containing no more than ***n*** propositions always terminates in time polynomial in ***n***.

ANS

From (b) we found that the total number of distinct 2CNF clauses is $2n^2 + 1$***.***
At any given step, the number of clauses we can generate through resolution is limited by the number of pairs of clauses that share complementary literals.
Now, total number of distinct resolutions would be $(2n^2 + 1)^2$***.***
Since each resolution step takes constant time (bounded by a polynomial in n), and there are at most $(2n^2 + 1)^2$ ***steps, because the number of clauses generated during the resolution process is bounded by 4^n, which is a polynomial function of n.***

d. Explain why your argument in (c) does not apply to **3-CNF**.

ANS

The reasoning that ensures resolution terminates in polynomial time applies to 2-CNF but not to 3-CNF because of the distinct ways clauses are structured. In 2-CNF, with just two literals per clause, it's manageable, whereas 3-CNF clauses contain exactly three literals each, which complicates the resolution process.
When resolving 3-CNF clauses, you need to consider triples of clauses, not just pairs, which significantly increases the number of potential combinations to examine during the resolution process.