Practice Problem Set 1 Solutions

1. **Unraveling the Maze**

Solution: In any sort of propositional encoding problem, we require some basic propositional variables. Here a natural choice is $p_{i,j,t}$, which evaluates to true if the agent is at position (i, j) at time step t.
Let $A = \{1, 2, ..., n\} \times \{1, 2, ..., n\}$ be the set of grid indices and $T = n^2$ be the maximum time steps.
We also append the grid to an $(n+1) \times (n+1)$ (0-indexed) grid, to handle out of bound access.
The complete formula is $F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$.

Valid Position (C_1)

At every time step $t \in \{0, \ldots, T\}$, the agent must occupy exactly one cell.

$$C_1 = \bigwedge_{t=0}^{T} \left(\left(\bigvee_{i,j \in A} p_{i,j,t} \right) \wedge \bigwedge_{(i,j) \neq (i',j')} (\neg p_{i,j,t} \vee \neg p_{i',j',t}) \right)$$

Walls and Boundaries (C_2)

The agent can never occupy a cell that is in the set of walls W.

$$C_2 = \bigwedge_{t=0}^{T} ((\bigwedge_{(i,j) \in W} \neg p_{i,j,t}) \wedge (\bigwedge_{i=0 \vee i=n+1 \vee j=0 \vee j=n+1} \neg p_{i,j,t}))$$

Start and End States (C_3)

The agent must start at the designated start point at $t = 0$ and reach the goal by $t = T$. To allow for paths shorter than T, we can specify that once the agent reaches the goal, they stay there.

$$C_3 = p_{i_0,j_0,0} \wedge p_{i_1,j_1,T}$$

Valid Transitions (C_4)

If the agent is at (i, j) at time t, at time $t + 1$ they must be in an adjacent cell. Let $Adj(i, j)$ be the set of 4-adjacent cells: $\{(i + 1, j), (i - 1, j), (i, j + 1), (i, j - 1)\}$. To allow the agent to "wait" at the goal, we include (i, j) in the allowed moves.

$$C_4 = \bigwedge_{t=0}^{T-1} \bigwedge_{i,j \in A} \left(p_{i,j,t} \implies \bigvee_{(i',j') \in Adj(i,j) \cup \{(i,j)\}} p_{i',j',t+1} \right)$$

2. **Theorem Encoding**

Solution: Let $P(i, j)$ represent the proposition that the i^{th} pigeon is sitting in the j^{th} hole, where $i \in \{1 \dots n+1\}$ and $j \in \{1 \dots n\}$.
The Pigeonhole principle states that, if there are $n+1$ pigeons and n holes, and every pigeon sits in exactly one hole, then there is a hole occupied by more than one pigeon. To convert this into a PL formula, let us convert each side of the implication into PL first.
Every pigeon sits in at least one hole can be expressed in PL as:

$$\bigwedge_{i=1}^{n+1} \bigvee_{j=1}^{n} P(i, j)$$

Here the inner disjunction refers to the i^{th} pigeon sitting in some hole, and the outer conjunction makes it so that every pigeon must sit in some hole. Call this condition F. We also need no pigeon to sit in multiple holes. Say pigeon i sits in holes j and k with $j < k$. The formula $P(i, j) \wedge P(i, k)$ represents this scenario. There exists a pigeon sitting in multiple holes therefore becomes:

$$\bigvee_{i=1}^{n+1} \bigvee_{\substack{j,k=1 \\ j<k}}^{n} (P(i, j) \wedge P(i, k))$$

Here, the inner disjunction refers to the i^{th} pigeon sitting in multiple holes and the outer disjunction refers to there existing a pigeon sitting in multiple holes.
Negating this, we get the condition for no pigeon to sit in multiple holes:

$$\bigwedge_{i=1}^{n+1} \bigwedge_{\substack{j,k=1 \\ j<k}}^{n} (\neg P(i, j) \vee \neg P(i, k))$$

Call this condition G.
Now, say hole k is occupied by pigeons i and j with $i < j$. We then have $P(i, k) \wedge P(j, k)$. There exists a hole occupied by more than one pigeon therefore becomes:

$$\bigvee_{k=1}^{n} \bigvee_{\substack{i,j=1 \\ i<j}}^{n+1} (P(i, k) \wedge P(j, k))$$

Here, the inner disjunction refers to the k^{th} hole being occupied by more than one pigeon and the outer disjunction refers to there existing a hole occupied by multiple pigeons. Call this condition H.
The Pigeonhole Principle therefore becomes:

$$F \wedge G \implies H$$

3. **Adequate Sets**

Solution:

Lemma. *For any formula made out of* $\{\top, \wedge, \vee, \rightarrow\}$*, setting all the propositional variables to 1 always results in the overall formula having a truth value of 1.*

Note that this immediately shows that no formula constructed only out of $\{\top, \wedge, \vee, \rightarrow\}$ can be equivalent to $\bot$.

We shall prove this lemma via structural induction.

Base Case:

If the formula just consists of a single propositional variable p or is just $\top$, the result clearly follows.

Inductive Hypothesis:

Say φ and ψ are formulae constructed only with $\{\top, \wedge, \vee, \rightarrow\}$ and setting all the propositional variables to 1 results in the truth values of both φ and ψ being 1.

Inductive Step: The formulae we can construct from φ and ψ are $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, $\top \wedge \varphi$, $\varphi \wedge \top$, $\top \vee \varphi$, $\varphi \vee \top$, $\top \rightarrow \varphi$, $\varphi \rightarrow \top$, and the last six formulas with φ replaced by ψ. If we set all propositional variables to 1, then by the inductive hypothesis, the truth value of φ and ψ also become 1, and it can be seen that the truth values of the new formulae are also 1.

Therefore, by structural induction, the lemma is proven and with it we have proved that $\{\top, \wedge, \vee, \rightarrow\}$ is inadequate.

4. **Who leaked it?**

Solution:

Variables:

- D: Dev is speaking the truth.
- E: Erin is speaking the truth.
- S: Sam is speaking the truth.
- p_1: Erin had the project folder open that evening.
- p_2: Erin was in the CC library that evening.

Interpretation of the statements :

$$D \longrightarrow p_1, \tag{1}$$
$$E \longrightarrow (\neg p_1 \wedge \neg p_2), \tag{2}$$
$$S \longrightarrow p_2. \tag{3}$$

Assumptions.

1. Exactly one student leaked the project. (Hence exactly two students are innocent.)
2. Every innocent student tells the truth; the culprit may lie. Therefore exactly two of D, E, S are true.
3. From the previous point we encode the "at least two speak truth" condition as:

$$(D \wedge E) \ \vee \ (E \wedge S) \ \vee \ (S \wedge D). \tag{4}$$

Case analysis. We test each disjunct in (4).

Case 1: $D \wedge E$ is true.
From (1) we get p_1. From (2) we get $\neg p_1$. Thus p_1 and $\neg p_1$ hold simultaneously, a contradiction. Hence the case $D \wedge E$ is impossible.

Case 2: $E \wedge S$ is true.
From (2) we get $\neg p_2$. From (3) we get p_2. Thus p_2 and $\neg p_2$ hold simultaneously, a contradiction. Hence the case $E \wedge S$ is impossible.

Case 3: $S \wedge D$ is true.
This is the only remaining disjunct in (4) after rejecting the first two. If $S \wedge D$ holds, then E must be false (Erin is not telling the truth). By our assumptions the culprit may lie while innocents tell the truth; since exactly one student is the culprit and E is the sole liar here, Erin is the culprit.

5. **Understanding properties of propositional formulas**

Solution:

1. True. Let σ be an arbitrary assignment. Since F is unsatisfiable, we have

$$[\![F]\!](\sigma) = 0$$

   Therefore, $[\![\neg F]\!](\sigma) = 1$. Since this holds for every assignment σ, it follows that $\neg F$ is valid.

2. False.
   A counter example is the formula $x \to \bot$, for any propositional variable x. We can find a satisfiable assignment for the whole formula (assignment for which x is 0), and also for x, but not for $\bot$.

3. True. Consider an assignment σ. If the value of P_1 under σ is false then outermost implication is true. If the value of P_1 under σ is true then arguing from inside outwards all the implication subformulas are true.

4. False
   If $S \models \bot$, then S is unsatisfiable. In this case, both $S \models F$ and $S \models \neg F$ hold, since the set of assignments that satisfy F is empty.

5. True
   Let σ be an assignment for which S is true. Since $S \models F \vee G$, either F or G is true for any such assignment. In the first case, since $S \cup \{F\} \models H$, H evaluates to true under σ. In second case since $S \cup \{G\} \models H$, H evaluates to true under σ. Since these two cases cover the only two possibilities, every assignment that evaluates S to true will evaluate H to true, so $S \models H$.

   You can also try to prove using natural deduction that

$$S \to F \vee G,\ S \wedge F \to H,\ S \wedge G \to H \ \vdash\ S \to H$$

   Since natural deduction is sound, everything that is syntactically entailed is also semantically entailed. Hence the given statement about semantic entailments hold.

6. **Tautologies in Propositional Logic**

(a) $\big((p \to q) \wedge (q \to r) \wedge (r \to s)\big) \to (p \to s)$.

Solution: Yes, it is a tautology. We verify this using a truth table. We leave it as an exercise to you to show using natural deduction that

$$\vdash \big((p \to q) \land (q \to r) \land (r \to s)\big) \to (p \to s)$$

Back to truth tables (certainly not the preferred way of giving an answer, but just so that you know how to systematically construct truth tables of small formulas). For readability, let:

$$A := p \to q, \quad B := q \to r, \quad C := r \to s, \quad D := A \land B \land C, \quad E := p \to s.$$

p	q	r	s	A	B	C	D	E	$D \to E$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$
$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\top$
$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$
$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\top$
$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$

Since the final column $(D \to E)$ is $\top$ for all 16 interpretations, the formula is a tautology.

(b) Let $p_1, \ldots, p_n, q$ be propositional variables with $n \geq 1$.

$$\Big(\bigwedge_{i=1}^{n}(p_i \to q)\Big) \to \Big((p_1 \lor p_2 \lor \cdots \lor p_n) \to q\Big).$$

Solution: Yes, it is a tautology.
A direct verification using a truth table is not feasible here, since the formula involves $n+1$ propositional variables and hence would require 2^{n+1} rows. We therefore reason directly using the semantics of propositional logic.
Let $\mathcal{I}$ be an arbitrary interpretation (some assignment of the variables). Suppose that the antecedent

$$\bigwedge_{i=1}^{n}(p_i \to q)$$

is true under $\mathcal{I}$. We must show that the consequent

$$(p_1 \lor p_2 \lor \cdots \lor p_n) \to q$$

is also true under $\mathcal{I}$.
There are two cases to consider.

- If $(p_1 \vee p_2 \vee \cdots \vee p_n)$ is false under $\mathcal{I}$, then the implication $(p_1 \vee p_2 \vee \cdots \vee p_n) \to q$ is true by the semantics of implication.
- If $(p_1 \vee p_2 \vee \cdots \vee p_n)$ is true under $\mathcal{I}$, then at least one of $p_1, \ldots, p_n$ must be true. Let p_k be such a variable. Since $p_k \to q$ is true under $\mathcal{I}$ and p_k itself is true, it follows that q is true under $\mathcal{I}$. Hence the implication $(p_1 \vee p_2 \vee \cdots \vee p_n) \to q$ is true in this case as well.

Thus, in all possible cases, the consequent is true whenever the antecedent is true. Since $\mathcal{I}$ was arbitrary, the formula is true under every interpretation and is therefore a tautology.

(c) Let $p_1, \ldots, p_n, q$ be propositional variables with $n \geq 1$.

$$\big((p_1 \wedge p_2 \wedge \cdots \wedge p_n) \to q\big) \leftrightarrow \big(p_n \to (p_{n-1} \to (\cdots \to (p_1 \to q)))\big).$$

Solution: Yes, it is a tautology. We prove this by induction on n.
Base case ($n = 1$): The formula reduces to

$$(p_1 \to q) \leftrightarrow (p_1 \to q),$$

which is trivially a tautology.
Inductive step: Assume that for some $n \geq 1$,

$$\big((p_1 \wedge \cdots \wedge p_n) \to q\big) \leftrightarrow \big(p_n \to (p_{n-1} \to (\cdots \to (p_1 \to q)))\big)$$

is a tautology.
We must show that

$$\big((p_1 \wedge \cdots \wedge p_n \wedge p_{n+1}) \to q\big) \leftrightarrow \big(p_{n+1} \to (p_n \to (\cdots \to (p_1 \to q)))\big)$$

is also a tautology.
Fix an arbitrary interpretation.

- If p_{n+1} is false, then the right-hand side is true by the semantics of implication. The left-hand side is also true since the antecedent $(p_1 \wedge \cdots \wedge p_n \wedge p_{n+1})$ is false.
- If p_{n+1} is true, then the right-hand side reduces to

  $$p_n \to (p_{n-1} \to (\cdots \to (p_1 \to q))),$$

  and the left-hand side reduces to

  $$(p_1 \wedge \cdots \wedge p_n) \to q.$$

  By the induction hypothesis, these two formulas have the same truth value.

Thus, in all cases, both sides of the biconditional have the same truth value. By induction, the equivalence holds for all $n \geq 1$, and hence the formula is a tautology.

7. **Incident Detection in a Feature Rollout System**

(a) **Solution:**

(a) The rules translate directly into the following set of formulas:

$$\Gamma_1 = \{\, (F \wedge E) \to I,\ I \to R,\ A \to \neg I \,\}.$$

(b) We check whether $\Gamma_1 \models (A \to \neg R)$ holds.

To show that the entailment does *not* hold, it suffices to give an interpretation that satisfies Γ_1 but falsifies $A \to \neg R$.

Consider an interpretation $\mathcal{I}$ under which:

$$A = \top, \quad I = \bot, \quad R = \top, \quad F = \bot, \quad E = \bot.$$

We verify that $\mathcal{I}$ satisfies all formulas in Γ_1:

- $(F \wedge E) \to I$ is true since $F \wedge E$ is false.
- $I \to R$ is true since I is false.
- $A \to \neg I$ is true since A is true and I is false.

Thus, $\mathcal{I}$ is a model of Γ_1. However, under $\mathcal{I}$, $A \to \neg R$ is false, since A is true and R is true.

Therefore,

$$\Gamma_1 \not\models (A \to \neg R).$$

(b) **Solution:** First note that Part (b) adds the two rules:

$$R \to I \qquad \text{and} \qquad R \to \neg F.$$

Thus,

$$\Gamma_2 = \{\, (F \wedge E) \to I,\ I \to R,\ A \to \neg I,\ R \to I,\ R \to \neg F \,\}.$$

(a) **Yes, Γ_2 is satisfiable.** For instance, the interpretation

$$F = \bot, \quad E = \bot, \quad I = \bot, \quad R = \bot, \quad A = \bot$$

satisfies every implication in Γ_2 (each has a false antecedent), so Γ_2 is satisfiable.

(b) **No.** We give a counter-interpretation.

Let $\mathcal{I}$ assign:

$$E = \top, \quad F = \bot, \quad I = \bot, \quad R = \bot, \quad A = \bot.$$

Then $(F \wedge E)$ is false, so $(F \wedge E) \to I$ is true. Also $I \to R$ is true since I is false. The formulas $A \to \neg I$, $R \to I$, and $R \to \neg F$ are all true since A and R are false. Thus $\mathcal{I}$ satisfies Γ_2.

However, under the same $\mathcal{I}$, $E \to I$ is false (since E is true and I is false). Therefore,

$$\Gamma_2 \not\models (E \to I).$$

(c) **Yes, $\Sigma_2 \models (I \leftrightarrow R)$.** Let $\mathcal{I}$ be any interpretation satisfying Σ_2. Since both $I \to R$ and $R \to I$ are in Σ_2, we have that I is true implies R is true, and R is true implies I is true. Hence, under every model of Σ_2, I and R have the same truth value. Therefore,

$$\Sigma_2 \models (I \leftrightarrow R).$$

8. **Course Selection**

Solution:

- We start with simple observations. For all i such that $1 \leq i \leq n$, we have

  $$x_i \leftrightarrow s_{i,1},$$

  and for all j such that $1 < j \leq k$,

  $$s_{1,j} = \bot.$$

- We can recursively observe that for all i such that $1 < i \leq n$ and $1 \leq j \leq k$,

  $$s_{i-1,j} \to s_{i,j}.$$

- For all i, j satisfying $1 < j \leq k$ and $1 < i \leq n$, we add the following constraints:

  $$(x_i \wedge s_{i-1,j-1}) \to s_{i,j}$$

- To enforce the final credit constraint $\sum_{i=1}^{n} x_i \leq k$, we must ensure that no additional course can be selected once k courses have already been chosen. This is captured by:

  $$x_i \to \neg s_{i-1,k} \quad \text{for all } i \text{ such that } 1 < i \leq n.$$

- The number of auxiliary variables $s_{i,j}$ is $O(nk)$, and each recursive step contributes only a constant number of clauses, yielding an overall encoding size of $O(nk)$ clauses. This encoding can be viewed as a propositional logic circuit similar to an adder or prefix-sum circuit studied in Digital Logic Design.

Note: For the constraint $\sum_{i=1}^{n} x_i \geq k$, there is an alternative encoding using $O(kn)$ auxiliary variables and $O(k^2 n)$ clauses based on the Pigeon-Hole Principle.

Let the variables x_i represent *courses*, and let $x_i = 1$ mean that course i is available to be taken. We introduce variables p_{ij} for $i \in [k]$ and $j \in [n]$, where $p_{ij} = 1$ means that the i-th required course slot is assigned to course j.

The constraints are:

$$\text{[available courses]} \quad \bigwedge_{i \in [k]} \bigwedge_{j \in [n]} (p_{ij} \to x_j)$$

$$\text{[at most one slot per course]} \quad \bigwedge_{\substack{i_1, i_2 \in [k] \\ i_1 \neq i_2}} \bigwedge_{j \in [n]} \neg (p_{i_1 j} \wedge p_{i_2 j})$$

[each slot assigned] $\bigwedge_{i\in[k]} \bigvee_{j\in[n]} p_{ij}$

Let f be the conjunction of all the above clauses. Then:

$$f \text{ is satisfiable } \leftrightarrow \text{ at least } k \text{ courses are selectable } \leftrightarrow \sum_{i=1}^{n} x_i \geq k.$$

9. **Doctor Strange's survival**

Solution:

- Thanos snaps if and only if he gets the Time Stone:
  $(Snap \leftrightarrow T)$
- If Iron Man does not survive, then Doctor Strange cannot survive.
  $(\neg I \rightarrow \neg S)$
- The Guardians help in the final battle only if Iron Man survives.
  $(G \rightarrow I)$
- Doctor Strange survives if and only if the snap happens and Iron Man survives.
  $\big(S \leftrightarrow (Snap \wedge I)\big)$

(a) Encoding:

$$(Snap \leftrightarrow T) \ \wedge \ (\neg I \rightarrow \neg S) \ \wedge \ (G \rightarrow I) \ \wedge \ \big(S \leftrightarrow (Snap \wedge I)\big)$$

(b) $S = true, T = true, I = true, Snap = true, G = false$ satisfies these constraints.
(c) $\neg I \rightarrow \neg S$ is implied by $S \leftrightarrow (Snap \wedge I)$.
(d) It is not implied. Can be seen from the constraints. Here is one counterexample, $G = true, I = true, S = false, Snap = false$ which is consistent with constraints. G is true and Snap is false possible. Thefore, $\neg(G \Rightarrow Snap)$.

10. **Graph connectivity**

Solution:

1. **Claim:** No negative literal will appear in an excellent formula g. First remove all clauses which contain both positive and negative literals of a propositional variable. Then if g has a negative literal, say $\neg p_{ij}$, then the formula g' obtained by removing all occurrences of $\neg p_{ij}$ in g is also an excellent formula.

   Proof:
   $\mathcal{A} \models g \implies \mathcal{A} \models g'$: Any assignment satisfying clause C would also satisfy $C \backslash \{\neg p_{ij}\}$. So any assignment satisfying g would also satisfy g'.

$\mathcal{A} \not\models g \implies \mathcal{A} \not\models g'$: On the other hand, for an assignment not satisfying g, the given graph G is not connected. If edge $v_i - v_j$ was not in G, then for any clause C in g, because C was not satisfied, $C \backslash \{\neg p_{ij}\}$ was also not satisfied. So g' is also not satisfied by the assignment.

If edge $v_i - v_j$ was present in G and G was not connected (g is not satisfied), then removing edge $v_i - v_j$ would still give a disconnected graph and hence an assignment not satisfying g. So for every clause C (in g) containing $\neg p_{ij}$, C is not satisfied by the modified assignment which implies $C \backslash \{\neg p_{ij}\}$ was not satisfied by the original assignment. The other clauses stay the same while moving from g to g', so g' is not satisfied by the original assignment.

So clearly, if there is an excellent formula with a negative literal, we can make another good formula with lesser terms, a contradiction.

2. For any tree T (having $n-1$ vertices), an excellent formula g must hold true. Also, removing any edge from T must give a graph which is disconnected and hence the corresponding assignment must not satisfy g. So there exists a clause in g with precisely the positive literals corresponding to the edges of T - no less, no more. This gives a clause for each tree which can be embedded in G. So a lower bound for the number of terms in an excellent formula is $(n-1) \times$ (No of trees which can be embedded in G)

3. **Claim:** The formula g_0 by taking the clauses with prop vars as edges of every tree possible in G as described in part 2 is a good formula.
Proof: Note that if an assignment satisfies a clause with only positive literals, adding an edge does not change the clause's satisfiability. Any connected graph has a spanning tree. So start with the assignment corresponding to a spanning tree of G. At least one clause C in the formula is satisfied. Now add edges to get to G. Satisfiability of C does not change. So assignment for a connected graph satisfies g_0.

   On the other hand, each clause in g_0 corresponds to edges inducing a tree with vertices in G. If G is not connected, no tree is induced by edges of G. Thus no clause is satisfied by assignment corresponding to G.

   So g_0 is the unique excellent formula.

4. By parts 2 and 3, the answer is $(n-1)\times$(No of trees which can be embedded in G). By Cayley's theorem (not part of this course, but just for your curiosity), this turns out to be $(n-1) \times n^{n-2}$

11. **Natural deduction**

Solution:

1. Structural induction. For a formula F in $\mathcal{P}$, let the corresponding formula in $\mathcal{Q}$ obtained using this method be denoted by $Q(F)$

2. Let the proof using original rules of natural deduction have formulae $F_1, F_2, \ldots F_n$ as its lines with respective justifications. Then the idea is to convert each of F_i to

their counterpart $Q(F_i)$ in $\mathcal{Q}$ and show that $Q(F_i)$ follows from $Q(F_1), Q(F_2), \ldots Q(F_{i-1})$ and $Q(\phi) = \phi$.

Then at the end, we will be left with $Q(\psi) = \psi$ and the proof will be complete.

To do this, for each proof rule in natural deduction, we have to find a way to execute that rule (with $Q(\cdot)$ of respective formulae) using only our new rules of natural deduction. E.g. $\neg\neg_e : \neg\neg p \vdash p$ can be written as $((p \to \bot) \to \bot) \vdash p$ which can be proved as:

(a) Assume $p \to \bot$

(b) $\bot$ $(\to_e)$

(c) p $(\bot_e)$

(d) Assumption finished

(e) $(p \to \bot) \to p$ $(\to_i)$

(f) Choose $\phi := p$, $\psi := (p \to \bot)$ and $\zeta := p$ in $(\to \bot)_e$ to get p

Do this for all proof rules and done! (beware! $\wedge_{e_1} : [(p \to (q \to \bot)) \to \bot] \vdash p$ is surprisingly non-trivial)

12. **Binary Decision Diagrams**

Solution: The truth table for the above BDD

x_1	x_2	x_3	ϕ
$\bot$	$\bot$	$\bot$	$\bot$
$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$
$\top$	$\bot$	$\top$	$\bot$
$\top$	$\top$	$\bot$	$\top$
$\top$	$\top$	$\top$	$\top$

Now, we create a Karnough Map of the above truth table:

$x_1x_2\backslash x_3$	0	1
00	0	1
01	0	1
11	1	1
10	0	0

Thus,

$$\phi = (\neg x_1 \vee x_2) \wedge (x_1 \vee x_3)$$

13. **Map Coloring of Indian States**

Solution: Let the propositional variables $S_{i,j}$ represent if state i has color j, i.e. $S_{i,j}$ is true iff state i is colored j, and the variables $a_{i,j}$ represent if state i is adjacent to state j.

(i)

$$\bigvee_{1 \leq j \leq k} S_{i,j} \quad \forall i, 1 \leq i \leq n$$

(ii)

$$\bigwedge_{1 \leq j,l \leq k, j \neq l} (\neg S_{i,j} \vee \neg S_{i,l}) \quad \forall i, 1 \leq i \leq n$$

(iii)

$$a_{i,j} \rightarrow \bigwedge_{1 \leq l \leq k} (\neg S_{i,l} \vee \neg S_{j,l}) \quad \forall i, j; i \neq j, 1 \leq i, j \leq n$$

Using these constraints the formula modelling k-colorable is:

$$\bigwedge_{1 \leq i \leq n} ((\bigvee_{1 \leq j \leq k} S_{i,j}) \wedge (\bigwedge_{1 \leq j,l \leq k, j \neq l} (\neg S_{i,j} \vee \neg S_{i,l}))) \wedge \bigwedge_{1 \leq i,j \leq n, i \neq j} (a_{i,j} \rightarrow \bigwedge_{1 \leq l \leq k} (\neg S_{i,l} \vee \neg S_{j,l}))$$