Exercise 2.1.  [Resolution Completeness]

1. Does $F \models C$ imply $F \vdash \text{Res } C$? Proof or counterexample!

2. Can you prove $F \models C$ by resolution?

Solution:
Resolution can be used to prove that $F \models \bot$. From the lecture notes: $F$ unsatisfiable iff $F \vdash \text{Res } \Box$.

1. Counterexample: $F = \{\}, C = \{\{A, \neg A\}\}$

2. $F \models C$
   
   iff $\models \neg F \lor C$
   
   iff $\neg F \lor C$ tautology
   
   iff $\neg(\neg F \lor C)$ unsatisfiable
   
   iff $\neg(\neg F \lor C) \vdash \text{Res } \Box$

Exercise 2.2.  [Resolution of Horn-Clauses]
Can the resolvent of two Horn-clauses be a non-Horn clause?

Solution:
No. Proof: Let $C_1, C_2$ be two Horn clauses. Both of them have at most one positive literal. Without loss of generality, let $A_i$ be the positive literal occurring in $C_1$. Hence, $\neg A_i$ occurs in $C_2$. From the Horn clause property, we get that there is no other positive literal in $C_1$ and at most one in $C_2$. The resolvent is $C' = (C_1 - \{A_i\}) \cup (C_2 - \{\neg A_i\})$. We count the positive literals: None in $(C_1 - \{A_i\})$ and at most one in $(C_2 - \{\neg A_i\})$. Hence, at most one positive literal in $C'$. 
Exercise 2.3.  [Optimizing Resolution]
We call a clause $C$ trivially true if $A_i \in C$ and $\neg A_i \in C$ for some atom $A_i$. Show that
the resolution algorithm remains complete if it does not consider trivially true clauses for resolution.

Solution:
Completeness: If $F$ unsatisfiable, then $F \vdash_{\text{Res'}} \square$.

First we prove a lemma: If $F$ is unsatisfiable and contains a trivially true clause $C$, then $F' = F - C$ is still unsatisfiable. Proof by contraposition. Assume $F - C$ is satisfiable. Because $C$ is trivially satisfiable, $(F - C) \cup C = F$ is satisfiable. It follows that we can construct a $F'$ that contains no trivial clauses.

Assume that $F$ is unsatisfiable. We modify the completeness proof of resolution. Recall that
that proof proceeds by induction on the number of atomic formulas in $F$. We strengthen the
induction by mandating that $F$ contains no trivially true clauses. Initially, this is guaranteed
by the lemma. If $F$ is an unsatisfiable set of clauses containing $n + 1$ atomic formulas, we
construct $F_0$ and $F_1$ by setting $A_{n+1}$ to 0 or 1, respectively. Both $F_0$ and $F_1$ are unsatisfiable.
Also, neither $F_0$ nor $F_1$ contain trivial clauses. By induction hypothesis, we can obtain
resolution proofs such that $F_0 \vdash_{\text{Res'}} \square$ and $F_1 \vdash_{\text{Res'}} \square$. Constructing the new resolution
proof for $F$ introduces no new trivial clauses.

Exercise 2.4.  [Finite Axiomatization]
Let $M_0$ and $M$ be sets of formulas. $M_0$ is called axiom schema for $M$, iff for all assignments
$A$: $A \models M_0$ iff $A \models M$.

A set $M$ is called finitely axiomatized iff there is a finite axiom schema for $M$.

1. Are all sets of formulas finitely axiomatized? Proof or counterexample!

2. Let $M = (F_i)_{i \in \mathbb{N}}$ be a sequence of formulas, such that for all $i$: $F_{i+1} \models F_i$, and not
$F_i \models F_{i+1}$. Is $M$ finitely axiomatized?

Solution:

1. Counterexample: $M = \{A_1, A_1 \land A_2, A_1 \land A_2 \land A_3, \ldots \}$. Assume there is a finite axiom
schema $M_0$. $M_0$ can only contain finitely many atoms. Let $A$ be an assignment that
maps all $A_i$ in $M_0$ to 1, but all other $A_i$ to 0. Hence, $A \models M_0$ but not $A \models M$.

2. The same counterexample as above works here.
Exercise 2.5. [Compactness Theorem]
Suppose every subset of $S$ is satisfiable. Show that then

\[
\begin{align*}
\text{every subset of } S \cup \{F\} \text{ is satisfiable or} \\
\text{every subset of } S \cup \{\neg F\} \text{ is satisfiable}
\end{align*}
\]

for any formula $F$.

Solution:
Proof by contradiction. Suppose $S \cup \{F\}$ has an unsatisfiable subset $M$ and $S \cup \{\neg F\}$ has an unsatisfiable subset $L$. We can assume that $M = M' \cup \{F\}$ and $L = L' \cup \{\neg F\}$ for some $M'$, $L'$ where $M' \subseteq S$ and $L' \subseteq S$ because every subset of $S$ is satisfiable. We additionally know that $M' \cup L'$ is satisfiable by assumption. Consider the sets

\[
M' \cup L' \cup \{F\} \quad \text{and} \quad M' \cup L' \cup \{\neg F\}
\]

Then one of them has to be satisfiable. (Let $A$ with $A \models M' \cup L'$. Then either $A \models F$ or $A \models \neg F$. That is, $A \models F$ or $A \models \neg F$.) This directly implies that either $M$ or $L$ is satisfiable, a contradiction.
Homework 2.1.  [Resolution]  (4 points)
Use the resolution procedure to decide if the following formulas are satisfiable. Show your work (by giving the corresponding DAG or linear derivation)!

1. \((A_1 \lor A_2 \lor \neg A_3) \land \neg A_1 \land (A_1 \lor A_2 \lor A_3) \land (A_1 \lor \neg A_2)\)
2. \((\neg A_1 \lor A_2) \land (\neg A_2 \lor A_3) \land (A_1 \lor \neg A_3) \land (A_1 \lor A_2 \lor A_3)\)

Homework 2.2.  [Negative Resolution]  (6 points)
We call a clause \(C\) negative if it only contains negative clauses. Show that resolution remains complete if it only resolves two clauses if one of them is negative.

Homework 2.3.  [Sequent Calculus]  (5 points)
Prove the following formula using a sequent calculus derivation.

\(( ((A_1 \rightarrow A_3) \rightarrow A_3) \rightarrow (A_1 \rightarrow A_2) \rightarrow (A_2 \rightarrow A_3) \rightarrow A_3 )\)

You may use Logitext\(^1\) to derive the tree. Note that Logitext uses a slightly different notation: ‘\(\vdash\)’ instead of ‘\(\implies\)’.

\(^1\)http://logitext.mit.edu/main
Exercise Sheet 2  Logics  Page 5

**Homework 2.4. [Application of the Compactness Theorem]**

A finitely branching tree has the following structure:

- There is exactly one root node.
- Every node has a finite number of children.

We assign the root node the *level* 0 and the children of a node at level $n$ the level $n + 1$. Let $T_n$ denote the set of all nodes at level $n$, and $T$ the set of all nodes, i.e. $T = \bigcup_{n \in \mathbb{N}} T_n$. Let $P_t$ for $t \in T$ be the set of parent nodes of a node, i.e. $t$ is a child (or grand-child, ...) of all $t' \in P_t$. A path is a sequence of connected nodes, starting from the root node.

Prove the following lemma using the compactness theorem: Every countably infinite, finitely branching tree has an infinite path.

*Hint:* Use the following template for the proof.

1. Fix a set of tree nodes $T$. This set is (countably) infinite. You can assume that the sets $T_n$ and the sets $P_t$ are given.

2. For each node $t \in T$, let $A_t$ be an atomic formula. If an assignment $\mathcal{A}$ makes $A_t$ true, the node $t$ is part of the path.

3. Define a set of propositions $S$ that together guarantee the existence of an infinite path. That set is composed of three subsets:
   
   (a) For each level $n \in \mathbb{N}$, a node $t \in T_n$ is part of the path.

   (b) If a node $t$ is part of the path, so are all of its parent nodes $t' \in P_t$.

   (c) For each level $n \in \mathbb{N}$, there is at most one node of level $n$ part of the path.

4. Show that any finite subset of $S' \subseteq S$ is satisfiable by constructing an assignment such that $\mathcal{A}_{S'} \models S'$. Consider the largest $n$ for which a proposition from subset (a) is contained in $S'$.

5. Hence, $S$ is satisfiable. Show that a model $\mathcal{A} \models S$ represents an infinite path in $T$. 