Construct your decomposition trees carefully. A wrong application of a decomposition rule results in \((0 \text{ pts})\) for the tree.

**QUESTION 1 (5pts)**

Consider a proof system \(RS'\) obtained from \(RS\) by changing the sequence \(\Gamma'\) into \(\Gamma\) in all of the rules of inference of \(RS\).

1. Prove that the rule \((\neg \cap)\) of \(RS'\) is strongly sound. You can use the shorthand notation.

   Solution
   
   Consider the rule \((\neg \cap)\).
   
   \[
   \begin{array}{c}
   \neg \cap \end{array}
   \begin{array}{c}
   \Gamma, \neg A, \neg B, \Delta
   
   \Gamma, \neg (A \cap B), \Delta
   \end{array}
   \]

   1. By the definition we have that

      \[
      v'(\Gamma, \neg A, \neg B, \Delta) = v'(\delta_{[\Gamma, \neg A, \neg B, \Delta]}) = v'(\Gamma) \cup v'(\neg A) \cup v'(\neg B) \cup v'(\Delta) = v'(\Gamma) \cup \neg v'(A) \cup \neg v'(B) \cup v'(\Delta)
      \]

      \[
      = v'(\Gamma) \cup \neg (v'(A) \cap v'(B)) \cup v'(\Delta) = v'(\Gamma) \cup v'(\neg (A \cap B)) \cup v'(\Delta) = v'(\delta_{[\Gamma, \neg (A \cap B), \Delta]}) = v'(\Gamma', \neg (A \cap B), \Delta)
      \]

      Shorthand Notation

      \[
      v'(\Gamma, \neg A, \neg B, \Delta) = \Gamma \cup (\neg A \cup \neg B) \cup \Delta = \Gamma \cup \neg (A \cap B) \cup \Delta = v'(\Gamma, \neg (A \cap B), \Delta)
      \]

      2. The proof of strong soundness of rules of \(RS'\) is obtained directly from corresponding proof in \(RS\) only by changing the sequence \(\Gamma'\) into \(\Gamma\).

**QUESTION 2 (5pts)**

Let \(GL\) be the Gentzen style proof system for classical logic.

1. Prove, by constructing a proper decomposition tree that

   \[
   \vdash_{GL}((\neg(a \cap b) \Rightarrow b) \Rightarrow (\neg b \Rightarrow (\neg a \cup \neg b))).
   \]

   Solution

   Consider the following tree.
T→A

\[\neg (a \land b) \Rightarrow b \Rightarrow (\neg b \Rightarrow (\neg a \lor \neg b))\]

\[\neg b, (\neg (a \land b) \Rightarrow b) \Rightarrow (\neg a \lor \neg b)\]

\[\neg b, (\neg (a \land b) \Rightarrow b) \Rightarrow \neg a, \neg b\]

\[\neg b, \neg (a \land b) \Rightarrow b \Rightarrow \neg a\]

\[\neg b, a, \neg b, (\neg (a \land b) \Rightarrow b) \Rightarrow b\]

\[b, a, (\neg (a \land b) \Rightarrow b) \Rightarrow b\]

\[ b, a \rightarrow \neg (a \land b), b \quad b, a, b \rightarrow b\]

\[ b, a, (a \land b) \rightarrow b\]

\[ b, a, a, b \rightarrow b\]

All leaves of the decomposition tree are axioms, hence the proof has been found.

2. Use the completeness theorem for GL to prove that

\[\neg_{GL} ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))\].

Solution

By the Completeness Theorem we have that

\[\neg_{GL} ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))\] if and only if \[\not\models ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))\]

Any v, such that v(a) = v(b) = F is a counter-model for ((a ⇒ b) ⇒ (¬b ⇒ a)), hence By the Completeness Theorem \[\neg_{GL} ((a ⇒ b) ⇒ (¬b ⇒ a))\]

QUESTION 3 (5pts) Let GL be the Gentzen style proof system for classical logic.

1. Define SHORTLY Decomposition Tree for any A in GL.
Solution

Here is my short definition.

**Decomposition Tree** $T_A$

For each formula $A \in F$, a decomposition tree $T_A$ is a tree built as follows.

**Step 1.** The sequent $\rightarrow A$ is the root of $T_A$ and for any node $\Gamma \rightarrow \Delta$ of the tree we follow the steps below.

**Step 2.** If $\Gamma \rightarrow \Delta$ is indecomposable, then $\Gamma \rightarrow \Delta$ becomes a leaf of the tree.

**Step 3.** If $\Gamma \rightarrow \Delta$ is decomposable, then we pick one rule that applies by matching the sequent of the current node with the domain of the rules. Then we apply this rule as decomposition rule and put its left and right premises as the left and right leaves, or as one leaf in case of one premiss rule.

**Step 4.** We repeat steps 2 and 3 until we obtain only indecomposable leaves.

2. Prove Completeness Theorem for **GL**. We assume that the STRONG soundness has been proved.

**Solution**

**Formula Completeness Theorem**

For any formula $A \in F$,

$\vdash_{GL} A$ if and only if $\models A$.

We prove the logically equivalent form of the Completeness part: for any $A \in F$

If $\not\vdash_{GL} A$ then $\not\models A$.

Assume $\not\vdash_{GL} A$, i.e. $\rightarrow A$ does not have a proof in **GL**. Let $T_A$ be a set of all decomposition trees of $\rightarrow A$. As $\not\vdash_{GL} A$, each $T \in T_A$ has a non-axiom leaf. We choose an arbitrary $T_A \in T_A$. Let $\Gamma' \rightarrow \Delta'$ be a non-axiom leaf of $T_A$, for $\Delta' \in VAR^*$ such that $\{\Gamma\} \cap \{\Delta\} = \emptyset$.

The non-axiom leaf $L = \Gamma' \rightarrow \Delta'$ defines a truth assignment $v : VAR \leftarrow \{T, F\}$ which falsifies $A$ as follows:

$$v(a) = \begin{cases} T & \text{if } a \text{ appears in } \Gamma' \\ F & \text{if } a \text{ appears in } \Delta' \\ \text{any value} & \text{if } a \text{ does not appear in } L \end{cases}$$

This proves, by strong soundness of the rules of inference of **GL** that $\not\models A$.

**QUESTION 4 (5pts)**

We know that a classical tautology $(\neg(a \land b) \cup (a \land b))$ is NOT Intuitionistic tautology and we know by **Tarski Theorem** that $\neg\neg((\neg(a \land b) \cup (a \land b))$ is intuitionistically PROVABLE

**FIND** the proof of the formula

$\neg\neg((\neg(a \land b) \cup (a \land b))$

in the Gentzen system **LI** for Intuitionistic Logic.

**Solution**
\[ \rightarrow \neg(\neg(a \cap b) \cup (a \cap b)) \]
\[ | (\rightarrow \neg) \]
\[ \neg(\neg(a \cap b) \cup (a \cap b)) \rightarrow \]
\[ | (contr \rightarrow) \]
\[ \neg(\neg(a \cap b) \cup (a \cap b)), \neg(\neg(a \cap b) \cup (a \cap b)) \rightarrow \]
\[ | (\neg \rightarrow) \]
\[ \neg(\neg(a \cap b) \cup (a \cap b)) \rightarrow (\neg(a \cap b) \cup (a \cap b)) \]
\[ | (\rightarrow \cup_1) \]
\[ \neg(\neg(a \cap b) \cup (a \cap b)) \rightarrow (a \cap b) \]
\[ | (\rightarrow \neg) \]
\[ (a \cap b), \neg(\neg(a \cap b) \cup (a \cap b)) \rightarrow \]
\[ | (exch \rightarrow) \]
\[ \neg(\neg(a \cap b) \cup (a \cap b)), (a \cap b), \rightarrow \]
\[ | (\neg \rightarrow) \]
\[ (a \cap b) \rightarrow (\neg(a \cap b) \cup (a \cap b)) \]
\[ | (\rightarrow \cup_2) \]
\[ (a \cap b) \rightarrow (a \cap b) \]
\[ axiom \]

**QUESTION 5 (10pts)**

Use the QRS proof system to prove that \( \not\models A \) for

\[ A = (\exists x(\neg P(x, y) \cap R(x)) \cap Q(x)) \Rightarrow \forall x((\neg P(x, y) \cap R(x)) \cap Q(x)) \]

where \( P \) is a two argument predicate symbol and \( R, Q \) one argument predicate symbols

1. (5pts) Build the Decomposition Tree \( T_A \).

You must write comments at each step of decomposition that uses the rules (\( \exists \)) and (\( \forall \)).

No comments (or wrong comments), or a wrong application of any rule results in (0 pts) for the tree.

2. (5pts) Define a counter model for \( A \) determined by a non-axiom leaf of the tree \( T_A \). Justify why it proves that \( \not\models A \).

**Solution**

1. (5pts)

The Decomposition Tree \( T_A \) is:
2. \(5\)pts \ Short Solution

Given the non-axiom leaf

\[
L_A = P(x_1, y), \neg R(x_1), \neg Q(x_1), Q(x_2)
\]

We define a structure \(M = [M, I]\) and an assignment \(v\), such that \((M, v) \not\models L_A\) as follows.

We take a the universe of \(M\) the set \(T\) of all terms of our language \(L\), i.e. we put \(M = T\).

We define the interpretation \(I\) as follows.

\(P_I(x_1, y)\) is false (does not hold) for \(x_1\) and for any \(y \in \text{VAR}\),

\(R_I(x_1)\) is true (holds) for \(x_1\) and \(Q_I(x_2)\) is false (does not hold) for \(x_2\), and
\(Q(x_i)\) is true (holds) for \(x_2\).

We define the assignment \(v: \text{VAR} \rightarrow T\) as identity, i.e., we put \(v(x) = x\) for any \(x \in \text{VAR}\).

**Longer Solution** you can add this for explanation

Obviously, for such defined structure \([M, I]\) and the assignment \(v\) we have that

\[
([T, I], v) \not\models P(x_1, y), \ ([T, I], v) \models R(x_1), \ ([T, I], v) \models Q(x_1), \text{ and } ([T, I], v) \not\models Q(x_2).
\]

We hence obtain that

\[
([T, I], v) \not\models P(x_1, y), \neg R(x_1), \neg Q(x_1), Q(x_2)
\]

This **proves** that such defined structure \([T, I]\) is a counter model for the non-axiom leaf

\[
L_A = P(x_1, y), \neg R(x_1), \neg Q(x_1), Q(x_2)
\]

3. (3pts)

By the **strong soundness** of QRS the structure \(M = [T, I]\) is also a counter-model for the formula \(A\), i.e. we proved that

\[
\not\models (\exists x((\neg P(x, y) \cap R(x)) \cap Q(x)) \Rightarrow \forall x((\neg (P(x, y) \cap R(x)) \cap Q(x))
\]

**REMARK 1**

We STOPED the decomposition process at the right branch of the \((\cap)\) rule on the node

\[
P(x_1, y), \neg R(x_1), \neg Q(x_1), ((\neg (P(x_2, y) \cap R(x_2)) \cap Q(x_2))
\]

If we decompose on the left branch of the \((\cap)\) rule we get other leaves as follows

\[
P(x_1, y), \neg R(x_1), \neg Q(x_1), (\neg (P(x_2, y) \cap R(x_2)))
\]

\[
\bigwedge \cap
\]

\[
P(x_1, y), \neg R(x_1), \neg Q(x_1), \neg P(x_2, y)
\]

\[
P(x_1, y), \neg R(x_1), \neg Q(x_1), R(x_2)
\]

\(x_1 \neq x_2, \text{Non-axiom}\)

**REMARK 2**

We define the counter models for the non-axiom leaves

\[
L_{1A} = P(x_1, y), \neg R(x_1), \neg Q(x_1), \neg P(x_2, y) \quad \text{and} \quad L_{2A} = P(x_1, y), \neg R(x_1), \neg Q(x_1), R(x_2)
\]

following a general definition below.

**Definition**

Given a **non-axiom leaf** \(L_A\) of a decomposition tree \(T_A\) we define a structure \(M = [M, I]\) and an assignment \(v\), such that \((M, v) \not\models L_A\) as follows.

We take a the universe of \(M\) the set \(T\) of all terms of our language \(L\), i.e. we put \(M = T\).
We define the interpretation $I$ as follows.

For any predicate symbol $Q \in P$, $\#Q = n$ we put that $Q(t_1, \ldots, t_n)$ is **true** (holds) for terms $t_1, \ldots, t_n$ if and only if the negation $\neg Q(t_1, \ldots, t_n)$ of the formula $Q(t_1, \ldots, t_n)$ appears on the leaf $L_A$ and we put $Q(t_1, \ldots, t_n)$ is **false** (does not hold) for terms $t_1, \ldots, t_n$ otherwise.

For any functional symbol $f \in F$, $\#f = n$ we put $f(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$. 