Problem 1 (6 points)

Write down a parenthesized version of each of the following boolean formulas according to the standard precedence rules of boolean algebra:

1. \( \neg P \lor Q \rightarrow P \land R \)
2. \( P \rightarrow Q \lor \neg R \land S \land P \rightarrow Q \)
3. \( A \lor B \land C \land \neg D \rightarrow A \)

Problem 2 (10 points)

For each of the following pairs of propositional formulas, determine if they are equivalent or not using the truth table method. Your solution should include (for each equivalence) the corresponding truth table and a brief sentence stating if the given equivalence is valid (or not) and why.

1. \( (A \rightarrow B) \land (A \lor B) \equiv B \)
2. \( (A \rightarrow B) \land (B \rightarrow C) \rightarrow (C \rightarrow A) \equiv A \)

Problem 3 (10 points)

Complete the following proof of the equivalence \( (\neg q \rightarrow r) \rightarrow r \equiv q \rightarrow r \) justifying each line with the name of the algebraic rule applied to it, and underlining the subformula the rule has been applied to. (As an example, the answer for the first step if provided.) You can use any rule from Theorem 2 from the textbook (page 6), as well as the rule \( p \rightarrow q \equiv \neg p \lor q \) {Definition of \( \rightarrow \)}.

\[
\begin{align*}
(\neg q \rightarrow r) \rightarrow r & \equiv \text{(definition of \( \rightarrow \))} \\
(\neg \neg q \lor r) \rightarrow r & \equiv \\
(q \lor r) \rightarrow r & \equiv \\
\neg (q \lor r) \lor r & \equiv \\
(\neg q \land \neg r) \lor r & \equiv \\
r \lor (\neg q \land \neg r) & \equiv \\
(r \lor \neg q) \land (r \lor \neg r) & \equiv \\
(r \lor \neg q) \land 1 & \equiv \\
r \lor \neg q & \equiv \\
\neg q \lor r & \equiv \\
q \rightarrow r & \equiv
\end{align*}
\]
Problem 4 (10 points)

Use the algebraic rules for boolean functions to prove that the equivalence 
\[ \neg((p \land q) \lor \neg(p \land q)) \equiv 0 \]
Your proof should be written similarly to the one in problem 3, writing the name of the algebraic rule applied at each step, and underlying the corresponding subformula.

Problem 5 (14 points)

Two components \( P_1, P_2 \) of a computer system need to access a shared resource that can serve only one component at a time.\(^1\) Access to the resource is governed by a controller as follows:

- The controller has two input lines \( x_1, x_2 \) and two output lines \( y_1, y_2 \) to communicate with the two components. (Each \( P_i \) can set the value of \( x_i \) and read the value of \( y_i \).)
- Each \( P_i \) requests access by setting \( x_i = 1 \). It then waits for \( y_i \) to take the value 1. Only then, it uses the resource. When \( P_i \) is done with using the resource, it releases it by setting \( x_i = 0 \).
- The controller grants or denies access to the resource by setting the value of \( y_1, y_2 \) as a function of the inputs \( x_1, x_2 \). Formally, the controller is described by two boolean expressions \( C_1(x_1, x_2) \) and \( C_2(x_1, x_2) \) which are used to determine the value of \( y_i = C_i(x_1, x_2) \) for \( i = 1, 2 \).

The controller should satisfy the following policies:

- (Mutual exclusion) \( P_1 \) and \( P_2 \) are never granted access to the resource at the same time.
- (Usefulness) If any of \( P_1 \) or \( P_2 \) requests the resource, then at least one of them is granted access to it by the controller.

As part of the solution to this problem, you should do the following:

1. Give a boolean formula \( M \) in the variables \( x_1, x_2, y_1, y_2 \) that describes the mutual exclusion property.
2. Give a boolean formula \( U \) in the variables \( x_1, x_2, y_1, y_2 \) that describes the usefulness property.
3. Give a formal description of a controller as a pair of boolean formulas \( C_1 \) and \( C_2 \) in the variables \( x_1, x_2 \). \( C_1 \) and \( C_2 \) should only make use of \( \neg, \land \) and \( \lor \) connectives, and should satisfy the given policy. Give also a brief English description of the ideas behind the design of your controller functions.
4. Show that your controller \( (C_1, C_2) \) satisfies mutual exclusion by proving\(^2\) that the compound formula \( M(x_1, x_2, C_1(x_1, x_2), C_2(x_1, x_2)) \) is valid, i.e., it is always true for any value of the inputs \( x_1, x_2 \).
5. Show that your controller \( (C_1, C_2) \) satisfies the usefulness property by proving that the compound formula \( U(x_1, x_2, C_1(x_1, x_2), C_2(x_1, x_2)) \) is valid, i.e., it is always true for any value of the inputs \( x_1, x_2 \).
6. Draw the diagram of a digital circuit corresponding to your controller functions \( C_1, C_2 \).

---

\(^1\)E.g., \( P_1 \) and \( P_2 \) may be two CPUs in a multiprocessor system that want to use a communication port.

\(^2\)You can prove 4 and 5 using either the truth table method, or the algebraic rules for Boolean formulas. (You can choose which method you like best or is more convenient.)