Problem 1

Will Owens has recently come to Yew Nork after graduating from Aspens Information Technology in Aspens, Greece. He's currently working at Gingko's paper and shipping company, and is studying circuitree\textsuperscript{1} in anticipation of an upcoming merger with the Pinary Tree Company.

Gingko’s circuits are outdated and the documentation has been stolen by Yew Nork’s notorious bootloggers. Will needs to create new circuits and document the languages of old ones, and he’s asked you to help him out.

a. Draw a circuit and a straight-line program for the Boolean function whose value is True exactly when \( x \) and \( y \) are True or \( z \) is False.

b. Determine the language of the function defined by the following circuit:

\textsuperscript{1}Circuitree, a branch of arboreal computation focused on modelling forest and logging ecosystems, is one of the most rapidly growing fields in computer science.
Problem 2

A set of Boolean functions Ω forms a complete basis if for every natural number \( n \), a logic circuit can be constructed for every Boolean function \( f : \{0, 1\}^n \rightarrow \{0, 1\} \) using only functions in Ω. For example, \{AND, OR, NOT\} is a complete basis.

a. Recall that a NAND gate is a gate on two inputs realizing an AND followed by a NOT function. That is, \( \text{NAND}(x, y) = \text{NOT}(\text{AND}(x, y)) \). Show that the basis consisting of only the NAND gate is a complete basis.

b. Determine whether or not the basis \{AND, OR\} is a complete basis. Justify your answer.

Problem 3

Will is working hard on his circuitree when he comes across a mysterious diagram for a finite-state machine. It was designed by the Pinary Tree Company and is labelled with the letters HP. Initials? A competing forestry company? A magical literary icon?

Unfortunately, Will doesn’t have time for curiosities. His Gingko’s bosses want all FSMs to be documented by state diagrams for tax purposes. To help him finish more quickly, convert the following circuit diagram for an FSM into an equivalent state diagram. Represent each state by a vertex and each transition by a directed edge.

In the above diagram, \( \oplus \) denotes Exclusive OR (XOR) and \( \wedge \) denotes AND.

Hint: Think of the memory bits as a single 2-bit binary integer.
The following questions are lab problems.

Lab Problem 1

In a dual-rail logic circuit, 0 and 1 are represented by the pairs \((0,1)\) and \((1,0)\), respectively. A variable \(x\) is represented by the pair \((x, \overline{x})\). The NOT function (called a DRL-NOT in this representation) is a pair of twisted wires.

a. How are DRL-AND and DRL-OR realized in this representation? Use standard AND and OR gates to construct circuits or expressions for gates in the new representation. Show that every function \(f : \{0,1\}^n \rightarrow \{0,1\}^m\) can be realized by a dual-rail logic circuit in which the standard NOT gates are used only on input variables (to obtain the pair \((x, \overline{x})\)).

b. Show that the dual-rail logic circuit for a function \(f : \{0,1\}^n \rightarrow \{0,1\}\) has size at most twice the circuit size and depth at most equal to the circuit depth of \(f\) over the basis \{AND, OR, NOT\}. Ignore the initial NOT gates to prepare the inputs.

Lab Problem 2

When Will goes home after a long day’s work, he walks by the gated house of his reclusive neighbor, U. K. Liptus. Will knows that U. K. Liptus is associated with the Discrete Logging Company, the main rival of the Pinary Tree Company. In fact, U. K. Liptus is so concerned with security that the property is gated with special dynamic AND gates. Depending on your security clearance, your key might open one, two, three, or many gates at a time. In order to reach the front door, you must open all of the gates.

Will realizes he can create a Boolean function to model opening the gates by having a variable \(x_i\) for each gate and an AND for the key clearance process. In particular, consider the Boolean function:

\[
f(x_1, x_2, \ldots, x_n) = x_1 \land x_2 \land \cdots \land x_n
\]

The number of gates required to construct a circuit that computes this function for a fixed \(n\) is dependent on the number of inputs allowed to each AND gate in the circuit (that is, the number of gates your key might open at a time).
a. Find an expression for the number of 2-input AND gates required to construct a circuit that computes \( f \), and an expression for the depth complexity of such a circuit.

b. Find expressions for the combinational complexity and depth complexity of a circuit constructed using \( r \)-input AND gates for an arbitrary (but fixed) \( r \).

**Lab Problem 3**

In class you saw two types of bit shifting:

- **Logical shifting**, where the bits are shifted and then open slots are filled with 0s. For example, logically shifting 10101110 left by two bits gives us 10111000.

- **Cyclic shifting**, where the bits that are shifted off one end wrap around to the other end. Cyclically shifting 10101110 left by two bits gives us 10111010.

a. Reduce logical shifting to cyclic shifting. That is, assume you have a machine \( M \) that can perform cyclic bit shifting. Construct a machine, using \( M \) as a subroutine, that performs logical bit shifting.

b. Reduce cyclic shifting to logical shifting.

**Note:** You may not explicitly inspect or change individual bits for either of these reductions. For part (a), you may assume that the number of bits to shift is less than the length of the 0/1 string.