Chapter 3 Problem Solutions

Samuel Lair

August 2022

Contents

1 Problem 1

If IMPLIES were redefined such that P IMPLIES Q is false when P is false (while keeping its definition the same when P is true), then this redefined IMPLIES would have the same truth table as AND.

2 Problem 2

2.1 (a)

$\neg Q \implies R$

2.2 (b)

$P \wedge Q \implies R$

2.3 (c)

$R \implies P$

2.4 (d)

$P \wedge \neg Q \implies R$

3 Problem 3

It is reasonable to translate these two IF-THEN statements in different ways into propositional formulas because the mother is invoking a conventional implication whereas the mathematician is invoking a mathematical implication.

In a conventional implication, there is a causal connection between hypothesis and conclusion. Either the hypothesis and conclusion are both true or they are both false. This corresponds to IFF.

A mathematical implication, however, lacks a casual connection between hypothesis and conclusion. If the hypothesis is true, the conclusion must be true as well. On the other hand, if the hypothesis is false, the conclusion can be either true or false. This corresponds to IMPLIES.

4 Problem 4

See ch3_p4.py for a Python script solution.
Sample output for n = 4:

```
T T T T
T T T F
T T F F
T T F T
T F T F
T F F F
T F F T
T F T T
F T T F
F T F F
F T F T
F F T F
F F F F
F F F T
F F T T
F T T T
```

5 Problem 5

We will denote truth assignments for formulas (*) and (**) as tuples of the form (P, Q, R).

5.1 (a)

P	Q	R	$(P \implies Q)$	$\wedge$	$(P \implies R)$	$\wedge$	$(R \implies P)$	$P \wedge$	$Q \wedge R$
T	T	T	T	**T**	T	T	T	**T**	T
T	T	F	T	**F**	F	F	T	**F**	F
T	F	T	F	**F**	T	T	T	**F**	F
T	F	F	F	**F**	F	F	T	**F**	F
F	T	T	T	**F**	T	F	F	**F**	T
F	T	F	T	**F**	F	F	T	**F**	F
F	F	T	T	**F**	T	F	F	**F**	F
F	F	F	T	**T**	T	T	T	**F**	F

(*) is T and (**) is F for (F, F, F).

5.2 (b)

Sloppy Sam is relying on conventional implication, where a hypothesis and a conclusion are either both true or both false, in a mathematical context, where the falsehood of a hypothesis does not affect the truth of a conclusion.

6 Problem 6

6.1 (a)

$$
\begin{aligned}
c_0 &= b \\
s_0 &= a_0 \oplus c_0 \\
c_1 &= a_0 \wedge c_0 \\
s_1 &= a_1 \oplus c_1 \\
c_2 &= a_1 \wedge c_1 \\
&\dots \\
c_i &= a_{i-1} \wedge c_{i-1} \\
s_i &= a_i \oplus c_1 \\
c_{i+1} &= a_i \wedge c_i \\
&\dots \\
c_n &= a_{n-1} \wedge c_{n-1} \\
s_n &= a_n \oplus c_n \\
c_{n+1} &= a_{n+1} \wedge c_{n+1} \\
c &= c_{n+1}
\end{aligned}
$$

I.e. $s_i = a_i \oplus c_i$ for $0 \leq i \leq n$, $c_0 = b$, and $c_i = a_{i-1} \wedge c_{i-1}$ for $1 \leq i \leq n+1$.

6.2 (b)

Notice how after the zeroth bit, we are chaining together 1-bit full adders. We could easily turn this into a $n+1$ bit full adder by introducing a carry bit

input and replacing the zeroth bit half=adder with a full adder.

$$
\begin{aligned}
c_0 &= b_0 \\
s_0 &= a_0 \oplus c_0 \\
c_1 &= a_0 \wedge c_0 \\
s_1 &= (a_1 \oplus b_1) \oplus c_0 \\
c_2 &= ((a_1 \oplus b_1) \wedge c_1) \vee (a_1 \wedge b_1) \\
&\ldots \\
c_i &= ((a_{i-1} \oplus b_{i-1}) \wedge c_{i-1}) \vee (a_{i-1} \wedge b_{i-1}) \\
s_i &= (a_i \oplus b_i) \oplus c_{i-1} \\
&\ldots \\
c_n &= ((a_{n-1} \oplus b_{n-1}) \wedge c_{n-1}) \vee (a_{n-1} \wedge b_{n-1}) \\
s_n &= (a_n \oplus b_n) \oplus c_{n-1} \\
c_{n+1} &= ((a_n \oplus b_n) \wedge c_n) \vee (a_n \wedge b_n) \\
c &= c_{n+1}
\end{aligned}
$$

I.e. $s_0 = a_0 \oplus c_0$, $s_i = (a_i \oplus b_i) \oplus c_{i-1}$ for $1 \leq i \leq n$, $c_0 = b_0$, $c_1 = a_0 \wedge c_0$, $c_i = ((a_{i-1} \oplus b_{i-1}) \wedge c_{i-1}) \vee (a_{i-1} \wedge b_{i-1})$ for $2 \leq i \leq n+1$.

6.3 (c)

$$
\begin{aligned}
\text{total } \wedge &= 2n+1 \\
\text{total } \vee &= n \\
\text{total } \oplus &= 3n+1
\end{aligned}
$$

7 Problem 7

7.1 (a)

$c = 1$ only if $a_i = 1 \; \forall i$ such that $0 \leq i \leq a_{2n+1}$. However, if this is the case, then both of the single size add1-modules carry. Hence:

$$c = c_{(1)} \wedge c_{(2)}$$

7.2 (b)

If the lower half does not carry, the upper half should remain unchanged. I.e. output of the upper half should be equal to its input. On the other hand, if the lower half does carry, the upper half should be incremented by 1. I.e. the output of the upper half should be equal to the output of the upper single-size add1=module. Hence:

$$p_{n+i} = (\neg c_{(1)} \wedge a_{n+i}) \vee (c_{(1)} \wedge r_{i-1})$$

7.3 (c)

If $b = 1$, we add 1 to a. Otherwise, a is unchanged:

$$s_i = (b \wedge p_i) \vee (\neg b \wedge a_i)$$

7.4 (c)

We found that there is 1 propositional operator for computing each output bit of a add-1 module. Furthermore, up to two more gates can be added for the upper bits when choosing between the output of the upper add-1 module and the raw input. I.e. the number of gates on the path from the input to output of any given bit of a add1-module is bounded by a constant: 3.

One may object by pointing out that the carry of the lower add-1 module must resolve before we can choose the output of the upper module. However, the power of the parallel half-adder is that we can grow it by adding add-1 modules of a constant size such that we only ever need to wait for one level of add-1 modules to resolve. The number of operators on the output path of the ripple carry half-adder, however, continues to increase. Hence, for each subsequent doubling of n, the latency of the parallel half adder remains nearly constant (but not quite; part (b) shows us that two gates are added to the maximum output path for each add-1 module added; the rate of growth is actually proportional to $\log_2 n$) for each doubling of n while latency of the ripple carry half-adder continues to double.

8 Problem 8

8.1 (a)

The formula in the claim has 6 propositional variables. Therefore, the corresponding truth table for this formula has $2^6 = 64$ rows.

8.2 (b)

Prove that there are exactly two truth environments for the variable M, N, P, Q, R, S that satisfy the following formula:

$$(\neq P \vee Q) \wedge (\neq Q \vee R) \wedge (\neg R \vee S) \wedge (\neg S \vee P) \wedge M \wedge \neg N \tag{1}$$

We will denote a truth a environment for (1) as a tuple of the form (M, N, P, Q, R, S).

Proof. The proof is by case analysis. Let us consider the following cases:

1. $P = T$
2. $P = F$

Case 1: In order for (1) to evaluate to true, each of the 6 clauses connected by $\wedge$ operators must evaluate to true. Since $P = T$, $(\neq P \vee Q) = T \implies Q = T$. Since $Q = T$, $(\neg Q \vee R) = T \implies R = T$. Since $R = T$, $(\neg R \vee S) = T \implies S = T$. Finally, $M = T \implies M = T$ and $\neg N = T \implies N = F$. Hence, in this case, (T, F, T, T, T, T) is the only environment under which (1) evaluates to T.

Case 2: In order for (1) to evaluate to true, each of the 6 clauses connected by $\wedge$ operators must evaluate to true. Since $P = F$, $(\neg S \vee P) = T \implies S = F$. Since $S = F$, $(\neg R \vee S) = T \implies R = F$. Since $R = F$, $(\neg Q \vee R) = T \implies Q = F$. Finally, $M = T \implies M = T$ and $\neg N = T \implies N = F$. Hence, in this case, (T, F, F, F, F, F) is the only environment under which (1) evaluates to T.

In Case 1, (T, F, T, T, T, T) is the only environment under which (1) evaluates to T. In Case 2, (T, F, F, F, F, F) is the only environment under which (1) evaluates to T. Since there are no further cases to consider, the claim must be true. □

9 Problem 9

9.1 (a)

$n - 1$

9.2 (b)

For each doubling of n, the maximum number of AND gates on the path from an input to the output doubles for the serial circuit but only increases by 1 for the tree circuit. Therefore, tree circuit is exponentially faster than the serial circuit.

9.3 (c)

Assume n is a power of two. Prove that the n-input tree circuit has $n - 1$ AND-gates.

Proof. The proof is by the WOP. Let C be the set of counterexamples to our claim. We will prove by contradiction that C is empty.

Suppose C is not empty. Then by the WOP $\exists k$ such that k is the smallest value of n for which our claim does not hold. We know that our claim holds for $n = 2$ since a 2-input tree circuit is simply a single AND-gate. We also know that our claim holds for $n = 4$ since Figure 3.4 depicts a 4-input tree circuit consisting of 3 AND-gates. Therefore, $k \geq 8$.

Since a k-input tree circuit is the smallest tree circuit for which our claim does not hold, our claim must hold for a $\frac{k}{2}$-input tree circuit. In other words, a $\frac{k}{2}$-input tree circuit has $\frac{k}{2} - 1$ AND-gates. In order to extend a $\frac{k}{2}$-input tree circuit to a k-input tree circuit, we must add a row of $\frac{k}{2}$ AND-gates. Therefore, a k-input tree circuit has $\frac{k}{2} - 1 + \frac{k}{2} = k - 1$ AND-gates.

We've reached a contradiction since we've shown that our claim holds for a k-input tree circuit. Hence, C is empty and our claim must be true. □

10 Problem 10

10.1 (a)

A formula is satisfiable $\iff$ its negation is not valid.

10.2 (b)

A formula is valid $\iff$ its negation is not satisfiable.

10.3 (c)

Formula F is equivalent to formula G $\iff$ $F \oplus G$ is not satisfiable.

11 Problem 11

We will denote environments for our propositional formulas as tuples of the form (M, P, Q). If a variable is not in a formula, it is omitted from the tuple.

11.1 (a)

$M \implies Q$ is satisfiable but not valid. E.g. this formula is T for (T, T) but F for (T, F).

11.2 (b)

$M \implies (\neg P \vee \neg Q)$ is satisfiable but not valid. E.g. this formula evaluates to T for (T, F, T) but F for (T, T, T).

11.3 (c)

$M \implies (M \wedge (P \implies M))$ is valid since it evaluates to T for all possible environments: (T, T), (T, F), (F, T), and (F, F).

11.4 (d)

$(P \vee Q) \implies Q$ is satisfiable but not valid. E.g. this formula evaluates to T for (T, T) but F for (T, F).

11.5 (e)

$(P \vee Q) \implies (\neg P \wedge \neg Q)$ is not satisfiable since it evaluates to F for all possible environments: (T, T), (T, F), (F, T), and (F, F).

11.6 (f)

$(P \vee Q) \implies (M \wedge (P \implies M))$ is satisfiable but not valid. E.g. this formula evaluates to T for (T, T, T) but F for (F, T, T)

11.7 (g)

$(P \oplus Q) \implies Q$ is satisfiable but not valid. E.g. this formula evaluates to T for (F, T) but to F for (T, F)

11.8 (h)

$(P \oplus Q) \implies (\neg P \vee \neg Q)$ is valid since it evaluates to T for all possible environments: (T, T), (T, F), (F, T), (F, F)

11.9 (i)

$(P \oplus Q) \implies (M \wedge (P \implies M))$ is satisfiable but not valid. E.g. this formula evaluates to T for (T, F, T) but F for (F, T, F)

12 Problem 12

Prove by truth table that:

$$(P \oplus Q) \equiv \neg(P \iff Q) \tag{2}$$

Proof.

P	Q	$(P \oplus Q)$	$\neg$	$(P \iff Q)$
T	T	**F**	**F**	T
T	F	**T**	**T**	F
F	T	**T**	**T**	F
F	F	**F**	**F**	T

The truth table for $(P \oplus Q)$ is identical to the truth table for $\neg(P \iff Q)$. Hence, (2) must be true. □

13 Problem 14

Prove by truth table that:

$$(P \vee (Q \wedge R)) \equiv ((P \vee Q) \wedge (P \vee R)) \tag{3}$$

Proof.

P	Q	R	$(P \vee$	$(Q \wedge R))$	$((P \vee Q)$	$\wedge$	$(P \vee R))$
T	T	T	**T**	T	T	**T**	T
T	T	F	**T**	F	T	**T**	T
T	F	T	**T**	F	T	**T**	T
T	F	F	**T**	F	T	**T**	T
F	T	T	**T**	T	T	**T**	T
F	T	F	**F**	F	T	**F**	F
F	F	T	**F**	F	F	**F**	T
F	F	F	**F**	F	F	**F**	F

The truth table for $(P \vee (Q \wedge R))$ is identical to the truth table for $((P \vee Q) \wedge (P \vee R))$. Hence, (3) must be true. □

14 Problem 15

14.1 (a)

Since this formula has 13 propositional variables, its *truth table* has $2^13 = 8192$ rows. It would be very hard for a single person to verify the validity of this formula by *truth table* using only pencil and paper.

14.2 (b)

Prove that

$$(\neg(\neg A \implies B) \wedge A \wedge C) \implies (D \wedge E \wedge F \wedge G \wedge H \wedge I \wedge J \wedge K \wedge L \wedge M) \text{ is valid} \quad (4)$$

Proof. The proof is by case analysis. Let us consider the cases:

1. $A = T$
2. $A = F$

Case 1: Since $A = T$, $(\neg A \implies B) = T$ and $(\neg(\neg A \implies B)) = F$. Therefore, $(\neg(\neg A \implies B) \wedge A \wedge C) = F$. Hence, (4) holds for this case.
Case 2: Since $A = F$, $(\neg(\neg A \implies B) \wedge A \wedge C) = F$. Hence, (4) holds for this case.
(4) holds for Cases 1 and 2. Hence, (4) must be true. □

15 Problem 16

15.1 (a)

Prove by truth table that

$$(P \implies Q) \vee (Q \implies P) \tag{5}$$

is valid.

Proof.

P	Q	$(P \implies Q)$	$\vee$	$(Q \implies P)$
T	T	T	**T**	T
T	F	F	**T**	T
F	T	T	**T**	F
F	F	T	**T**	T

(5) evaluates to True in all of the rows of its truth table. Hence, (5) is valid. □

15.2 (b)

$$R ::= (P \wedge Q) \vee (\neg P \wedge \neg Q)$$

I.e. $R \equiv P \iff Q$.

15.3 (c)

Explain why

$$Q ::= P \text{ is valid} \iff \neg P$$

is not satisfiable.

$A \iff B$ is True only when A and B are both True or when A and B are both false. If P is valid is True, then $\neg P$ is always False. Hence, Q is not satisfiable.

15.4 (d)

Let

$$S ::= \neg P_1 \vee \neg P_2 \vee ...\neg P_k$$

S is True iff at least one of the P_i's are False. Hence, $P_1, ..., P_k$ is not consistent iff S is valid.

16 Problem 17

16.1 (a)

The specification is described by the following system of propositional formulas:

$$\neg L \implies Q \tag{6}$$

$$\neg L \implies B \tag{7}$$

$$\neg L \iff N \tag{8}$$

$$\neg Q \implies B \tag{9}$$

$$\neg B \tag{10}$$

16.2 (b)

We will denote a truth assignment for the specification as a tuple (L, Q, B, N).

In order to satisfy this specification, all five equations in our system of equations must evaluate to T:

$$\begin{aligned}
((10) = T) &\implies (B = F) \\
((B = F) \wedge ((9) = T)) &\implies (Q = T) \\
((B = F) \wedge ((7) = T)) &\implies (L = T) \\
((L = T) \wedge ((8) = T)) &\implies (N = F) \\
((L = T) \wedge (Q = T)) &\implies ((6) = T)
\end{aligned}$$

Therefore, the truth assignment (T, T, F, F) satisfies the specification. Hence, the specification is satisfiable.

16.3 (c)

The truth assignment in part(b) is the only one that satisfies the specification. ((10) = T) requires that $B = F$. This starts a chain of implications via the other equations that locks the rest of the propositional variables to a particular value. See the reasoning presented in part(b) for further details.

17 Problem 18

17.1 (a)

$$\begin{aligned} A \text{ IMPLIES } B &\equiv \text{NOT}(A) \text{ OR } B \\ A \text{ IFF } B &\equiv (A \text{ AND } B) \text{ OR } (\text{NOT}(A) \text{ AND NOT}(B)) \\ A \text{ XOR } B &\equiv (A \text{ OR } B) \text{ AND NOT}(A \text{ AND } B) \end{aligned}$$

Therefore, every propositional operator is equivalent to some combination of AND-OR-NOT operators. Hence, every propositional formula is equivalent to an AND-OR-NOT formula.

17.2 (b)

From (3.15), De Morgan's Law for OR, we can conclude that:

$$A \text{ AND } B \equiv \text{NOT}(\text{NOT}(A) \text{ OR NOT}(B))$$

Hence, we don't even need AND since it is equivalent to a combination of OR-NOT operators.

17.3 (c)

Observe that:

$$\begin{aligned} A \text{ NAND } B &::= \text{NOT}(A \text{ AND } B) \\ A \text{ AND } B &\equiv \text{NOT}(A \text{ NAND } B) \\ \text{NOT}(A) &\equiv A \text{ NAND } A \\ A \text{ AND } B &\equiv (A \text{ NAND } B) \text{ NAND } (A \text{ NAND } B) \end{aligned}$$

From (3.14), De Morgan's Law for AND, we can conclude that:

$$\begin{aligned} A \text{ OR } B &\equiv \text{NOT}(\text{NOT}(A) \text{ AND NOT}(B)) \\ A \text{ OR } B &\equiv (A \text{ NAND } A) \text{ NAND } (B \text{ NAND } B) \end{aligned}$$

Since AND-OR-NOT operators are equivalent to some combination of NAND operators and every propositional operator is equivalent to some combination of AND-OR-NOT operators, every propositional operator is equivalent to some combination of NAND operators. Hence, every propositional formula is equivalent to a NAND formula. I.e. we can "get by" with just NAND.

18 Problem 19

$$P ::= (A \wedge B \wedge \neg C \wedge D \wedge \neg E) \vee (\neg A \wedge B \wedge \neg C \wedge \neg E)$$

18.1 (a)

P is almost in Full Disjunctive Normal Form. The only issue is that the second clause doesn't mention D. We can introduce it by applying Validity for OR and identity for AND:

$$(\neg A \wedge B \wedge \neg C \wedge \neg E) \equiv (\neg A \wedge B \wedge \neg C \wedge \neg E) \wedge (D \vee \neg D)$$

Now apply distributivity of OR over AND:

$$\begin{aligned}(\neg A \wedge B \wedge \neg C \wedge \neg E) \equiv &(\neg A \wedge B \wedge \neg C \wedge D \wedge \neg E) \vee \\ &(\neg A \wedge B \wedge \neg C \wedge \neg D \wedge \neg E)\end{aligned}$$

We can now write a Full Disjunctive Normal Form for P:

$$\begin{aligned}P \equiv &(A \wedge B \wedge \neg C \wedge D \wedge \neg E) \vee \\ &(\neg A \wedge B \wedge \neg C \wedge D \wedge \neg E) \vee \\ &(\neg A \wedge B \wedge \neg C \wedge \neg D \wedge \neg E)\end{aligned} \tag{10}$$

18.2 (b)

I would expect C to have 29 clauses.

One way to derive the FDN and FCN forms of a propositional formula is to inspect its truth table. Each row where the propositional formula evaluates to true corresponds to a clause in its FDNF. Each row where the propositional formula evaluates to false corresponds to a clause in its FCNF. Since P has 5 distinct propositional variables, its truth table has $2^5 = 32$ rows. (10) has 3 clauses so P evaluates to true in only 3 out of 32 rows of its truth table. Therefore, P must evaluate to false in the other 29 rows of its truth table. Hence, P's FCNF has 29 clauses.

19 Problem 20

In sections (a) and (b) of Problem 18, we demonstrated that every propositional operator is equivalent to some combination of OR-NOT operators. Therefore, every propositional formula is equivalent to some formula whose only connectives are OR-NOT operators.

This problem is the NOR analog to section (c) of Problem 18. Observe that:

$$\begin{aligned}
&\text{NOT}(A) \equiv A \text{ NOR } A \\
&A \text{ OR } B \equiv \text{NOT}(A \text{ NOR } B) \\
&A \text{ OR } B \equiv (A \text{ NOR } B) \text{ NOR } (A \text{ NOR } B)
\end{aligned}$$

Therefore, both NOT and OR are equivalent to some combination of NOR operators. Hence, every propositional formula is equivalent to some formula whose only connectives are NOR operators because every propositional formula is equivalent to some formula whose only connectives are OR-NOT operators.

20 Problem 21

Let us denote our propositional formula as P. Its complement is $\neg P$. Suppose we have a DNF for $\neg P$:

$$\neg P \equiv Q_1 \vee Q_2 \vee ... \vee Q_n \tag{11}$$

where each Q_i is a conjunction of propositional variables in P (or their negations). Applying (3.15), De Morgan's Law for OR, to (11):

$$P \equiv \neg Q_1 \wedge \neg Q_2 \wedge ... \wedge \neg Q_n \tag{12}$$

Furthermore, from (3.14), De Morgan's Law for AND, we know that the negation of a conjunction is equivalent to a disjunction. Therefore, $\forall i \in \mathbb{Z}^+$ such that $1 \leq i \leq n$, $\neg Q_i \equiv R_i$ where each R_i is a disjunction of propositional variables in P (or their negations). I.e. by applying (3.14) to each $\neg Q_i$ in (12), we find:

$$P \equiv R_1 \wedge R_2 \wedge ... \wedge R_n \tag{13}$$

Hence, (13) is a CNF for P.

21 Problem 22

Disjunctive Normal Form:

$$\begin{aligned} P \equiv &(A \wedge B \wedge C \wedge D) \vee \\ &(A \wedge B \wedge \neg C \wedge D) \vee \\ &(A \wedge \neg B \wedge C \wedge D) \vee \\ &(A \wedge \neg B \wedge C \wedge \neg D) \vee \\ &(A \wedge \neg B \wedge \neg C \wedge D) \vee \\ &(A \wedge \neg B \wedge \neg C \wedge \neg D) \vee \\ &(\neg A \wedge B \wedge C \wedge D) \vee \\ &(\neg A \wedge B \wedge \neg C \wedge D) \vee \\ &(\neg A \wedge \neg B \wedge \neg C \wedge D) \vee \\ &(\neg A \wedge \neg B \wedge \neg C \wedge \neg D) \end{aligned}$$

Conjunctive Normal Form:

$$\begin{aligned} P \equiv &(\neg A \vee \neg B \vee \neg C \vee D) \wedge \\ &(\neg A \vee \neg B \vee C \vee D) \wedge \\ &(A \vee \neg B \vee \neg C \vee D) \\ &(A \vee \neg B \vee C \vee D) \\ &(A \vee B \vee \neg C \vee \neg D) \\ &(A \vee B \vee \neg C \vee D) \end{aligned}$$

22 Problem 24

22.1 (a)

For every binary gate $G_i \in C$, construct a constraint formula as follows:

$$X_i ::= (P_i \wedge Q_i) \iff R_i$$

where P_i and Q_i are the input wire variables of G_i and R_i is the output wire variable of G_i. n is the number of binary gates in C and G_n is chosen such that its output wire variable R_n corresponds to the output of C. Now let:

$$Y ::= X_1 \wedge X_2 \wedge ... \wedge X_n$$
$$F_C ::= Y \wedge R_n$$

The conjunction of constraints Y ensures that F_C is True only if R_n is equal to the output of C and the final conjunction with R_n ensures that F_C is True only if R_n is True. Hence, F_C is satisfiable iff C gives output T for some set of input values.

22.2 (b)

Since the size of F_C is proportional to the number of wires in C, an efficient method of solving F_C's satisfiability is also an efficient method of solving C's satisfiability. Hence, we can conclude that an efficient way of solving SAT would yield an efficient way to solve circuit-SAT.

23 Problem 25

23.1 (a)

Consider:

$$\begin{aligned}&(X_1 \iff (P \oplus Q))\wedge\\ &(X_2 \iff (X_1 \oplus R))\wedge\\ &(A \iff (\neg P \wedge S))\wedge\\ &(O \iff (X_2 \vee A))\end{aligned} \tag{13}$$

$$(13) \wedge O \tag{14}$$

(13) evaluates to True iff O evaluates to the same value as (3.33). Therefore, (14) evaluates to True iff O evaluates to the same value as (3.33) and O is True. Hence, (14) is satisfiable iff 3.33 is satisfiable.

23.2 (b)

Each constraining formula has 3 propositional variables. Therefore, the truth table for each constraining formula has $2^3 = 8$ rows. The FCNF of a propositional formula has a clause for each row of the truth table that evaluates to False, and each clause has one occurrence of each propositional variable. Hence, each constraining formula is equivalent to a 3CNF formula with at most $8 * 3 = 24$ occurrences of variables.

23.3 (c)

Create a propositional variable X_i for each propositional operator in F. Then for each propositional variable, create a constraining formula of the form $X_i \iff operand_{i1}\ operator_i\ operator_{i2}$. The operands depend on what the operator acts on in F. They could be one of the propositional variables in F or the output of another operator. In the latter case, the operand will be X_j for some $j \neq i$. Next, identify the X_k that does not appear on the right side of any of the constraining formulas. Its value will be equal to F when all of the constraints evaluate to true. Finally, we construct $C(F)$ by "anding" all of the constraints with each other and with X_k.

24 Problem 26

24.1 (a)

(i), (iii)

24.2 (b)

(i), (iii), (iv)

24.3 (c)

(i), (ii), (iv)

25 Problem 27

(a), (d)

26 Problem 28

26.1 (a)

$$(\forall x)(P(x) \implies \neg S(x))$$

26.2 (b)

$$(\forall x)(R(x) \implies S(x))$$

Alternately:

$$(\nexists x)(R(x) \implies \neg S(x))$$

26.3 (c)

$$(\forall x)(Q(x) \implies P(x))$$

26.4 (d)

$$(\forall x)(Q(x) \implies \neg R(x))$$

Alternatively:

$$(\nexists x)(Q(x) \implies R(x))$$

26.5 (e)

Part (d) follows logically from parts (a), (b), and (c).

Proof.

$$\begin{array}{ll} (\forall x)(Q(x) \implies P(x)) & (\text{apply (c)}) \implies \\ (\forall x)(Q(x) \implies \neg S(x)) & (\text{apply (a)}) \implies \\ (\forall x)(Q(x) \implies \neg R(x)) & (\text{apply (b)}) \end{array}$$

□

26.6 (f)

Everyone who comes from the 23rd century or likes to eat pizza is a 6.042 TA.

26.7 (g)

If there is someone from the 23rd century who isn't a 6.042 TA, then all monkeys like to eat pizza.

27 Problem 29

Show that

$$(\forall x \exists y.P(x,y)) \implies \forall z.P(z,z) \tag{15}$$

is not valid by describing a counter-model.

Let the nonnegative integers be our domain of discourse and

$$P(x,y) ::= x < y$$

P(x, y) is always True since if $x \in mathbbN$ then $(x+1) \in mathbbN$ and $x < (x+1)$. However, P(Z) is never True since if $z \in mathbbN$, $z = z$. It follows that (15) is False under this model. Hence, (15) is not valid.

28 Problem 30

Find a counter-model showing that the following is not valid:

$$\exists x.P(x) \implies \forall x.P(x) \tag{16}$$

Let the nonnegative integers be our domain of discourse and

$$P(x) ::= x = 1$$

$1 \in \mathbb{N}$ and $P(1) = True$. However, $2 \in \mathbb{N}$ and $P(2) = False$. It follows that (16) is False under this model. Hence, (16) is not valid.

29 Problem 31

Find a counter-model showing that the following is not valid:

$$[\exists x.P(x) \wedge \exists x.Q(x)] \implies \exists x.[P(x) \wedge Q(x)] \tag{17}$$

Let the nonnegative integers be our domain of discourse and

$$P(x) = x \geq 10$$
$$Q(x) = x < 10$$

$10 \in \mathbb{N}$ and $P(10) = True$. $9 \in \mathbb{N}$ and $Q(9) = True$. However, there is no $x \in \mathbb{N}$ such that both P(x) and Q(x) are True. It follows that (17) is False under this model. Hence, (17) is valid.

30 Problem 32

30.1 (a)

valid

30.2 (b)

invalid Let the nonnegative integers be our domain of discourse and

$$Q(x, y) ::= x < y$$

30.3 (c)

valid

30.4 (d)

valid

31 Problem 33

31.1 (a)

Prove that

$$(P \implies Q) \vee (Q \implies P) \tag{18}$$

is valid.

Proof. The proof is by truth table.

P	Q	$(P \implies Q)$	$\vee$	$(Q \implies P)$
T	T	T	**T**	T
T	F	F	**T**	T
F	T	T	**T**	F
F	F	T	**T**	T

(18) evaluates to True in all of the rows of its truth table. Hence, (18) is valid. □

31.2 (b)

This is another fallacious argument that fails to take into account that, unlike conventional implication, there is no causal relationship between hypothesis and conclusion in mathematical implication. Let F be a function and

$$P ::= F \text{ is differentiable}$$
$$Q ::= F \text{ is not continuous}$$

If F is a discontinuous differential function, then both P and Q are True and the first row of our truth table shows us, unsurprisingly, that (18) is True as well.

If F is a continuous differential function, then P is True but Q is False. The second row of our truth table shows us that $P \implies Q$ is False. However, $Q \implies P$ is True due to the lack of a causal relationship between hypothesis and conclusion in mathematical implication. Therefore, (18) is True.

If F is a discontinuous non-differentiable function, then P is False but Q is True. The third row of our truth table shows us that $Q \implies P$ is False. However, $Q \implies P$ is True due to the lack of a causal relationship between hypothesis and conclusion in mathematical implication. Therefore, (18) is True.

If F is a continuous non-differential function, then both P and Q are false. However, due to the lack of a causal relationship between hypothesis and conclusion in mathematical implication, $P \implies Q$ and $Q \implies P$ are both True. Therefore, (18) is True.

Our "sound" proof method is, in fact, fallacious due to the lack of a causal relationship between hypothesis and conclusion in mathematical implication. Whether a proposition is True or False tells us nothing about whether its converse is True or False.

32 Problem 35

32.1 (a)

$$\exists x.x^2 = 2$$

$\sqrt{2}$ is irrational.
False when the domain of discourse is $\mathbb{N}$.
False when the domain of discourse is $\mathbb{Z}$.
False when the domain of discourse is $\mathbb{Q}$.
True when the domain of discourse is $\mathbb{R}$.
True when the domain of discourse is $\mathbb{C}$.

32.2 (b)

$$\forall x.\exists y.x^2 = y$$

True when the domain of discourse is $\mathbb{N}$.
True when the domain of discourse is $\mathbb{Z}$.
True when the domain of discourse is $\mathbb{Q}$.
True when the domain of discourse is $\mathbb{R}$.
True when the domain of discourse is $\mathbb{C}$.

32.3 (c)

$$\forall y.\exists x.x^2 = y$$

This is a generalization of part(a).
False when the domain of discourse is $\mathbb{N}$.
False when the domain of discourse is $\mathbb{Z}$.
False when the domain of discourse is $\mathbb{Q}$.
False when the domain of discourse is $\mathbb{R}$. We need to consider the case when $y < 0$.
True when the domain of discourse is $\mathbb{C}$.

32.4 (d)

$$\forall x \neq 0.\exists y.xy = 1$$

For each $x \neq 0$ in a domain of discourse, determine whether $\frac{1}{x}$ is also in said domain of discourse.
False when the domain of discourse is $\mathbb{N}$.
False when the domain of discourse is $\mathbb{Z}$.
True when the domain of discourse is $\mathbb{Q}$.
True when the domain of discourse is $\mathbb{R}$.
True when the domain of discourse is $\mathbb{C}$.

32.5 (e)

$$\exists x.\exists y.x + 2y = 2 \wedge 2x + 4y = 5$$

If you multiply the left-hand equation by 2 and subtract from the right-hand equation, you get $0 = 1$ which is clearly not True. Therefore, this system of equations doesn't have a solution. Otherwise, one would solve the system of equations and determine which domains of discourse contain the solution.

False when the domain of discourse is $\mathbb{N}$.
False when the domain of discourse is $\mathbb{Z}$.
False when the domain of discourse is $\mathbb{Q}$.
False when the domain of discourse is $\mathbb{R}$.
False when the domain of discourse is $\mathbb{C}$.

33 Problem 36

33.1 (a)

$$\exists y.(x = yyy)$$

33.2 (b)

$$\text{NO-1S}(x) \wedge \exists y.(x = yy)$$

33.3 (c)

$$\text{NO-1S}(x) \vee \neg\text{SUBSTRING}(0, x)$$

33.4 (d)

$$x = 1 \vee \exists y.(x = 1y1 \wedge \text{NO-1S}(y))$$

33.5 (e)

Prepending a 0 to x shifts all of the digits of x one place to the right. If x is empty or a string of 0's, then x will obviously be a prefix of $0x$. However, if x contains any 1's, the first 1 will be shifted space to the right in $0x$ such that x will not be a prefix of $0x$. Hence, (*) is true only when x is a string of 0's.

34 Problem 37

34.1 (a)

$$Equal(m,n) ::= A(2n,n,m)$$

34.2 (b)

$$One(n) ::= M(n,n,n)$$

34.3 (c)

$$M(n,i,x) \land A(x,y,z) \land M(y,m,j) \land M(z,k,k)$$

34.4 (d)

$$Prime(p) ::= Greater(p,1) \land \forall x \forall y.(M(p,x,y) \implies (One(x) \land Equal(y,p)) \lor (One(y) \land Equal(x,p)))$$

34.5 (e)

$$Two(n) ::= M(2n,n,n)$$

34.6 (f)

$$Even(n) ::= \exists x \exists y.(Two(x) \land \neg Zero(y) \land M(n,x,y))$$

34.7 (g)

$$\forall n.(Even(n) \land \neg Two(n) \implies \exists x \exists y.(Prime(x) \land Prime(y) \land A(n,x,y)))$$

34.8 (h)

$$\forall n.(Greater(n,2) \implies \neg(\exists a \exists b \exists c.(X(a,x,n) \land X(b,y,n) \land X(c,z,n) \land A(c,a,b))))$$

34.9 (i)

$$\forall n \exists p.(Greater(p,n) \land Prime(p) \land \exists q.(Prime(q) \land A(p,q,2)))$$

35 Problem 38

35.1 (a)

$$m|n ::= \exists k.(k \neq 0 \wedge km = n)$$

35.2 (b)

$$P(n) ::= n \neq 0 \wedge n \neq 1 \wedge \nexists k.(k \neq 1 \wedge k \neq n \wedge k|n)$$

35.3 (c)

$$n \neq 0 \wedge (n = 1 \vee \exists p.(P(p) \wedge \forall k.(P(k) \wedge k|n \implies k = p)))$$

36 Problem 39

$$(\exists a \exists b \exists c \forall d.(d \neq a \wedge d \neq b \wedge d \neq c)) \implies \neg E(a, d)$$