HOMEWORK 01, DUE BY 2022-01-13

Question 1: Consider the following cubic equation: $x^3 + 2x^2 - 4x - 3 = 0$.

(A) It is known that this equation has an integer root r. Guess that root. Divide the cubic polynomial with $(x - r)$, where r is replaced by the root you guessed.

(B) Find all roots of this cubic equation.

(C) Factorize the polynomial $P(x) = x^3 + 2x^2 - 4x - 3$: Express it as a product $(x - a)(x - b)(x - c)$, where a, b, c are real numbers.

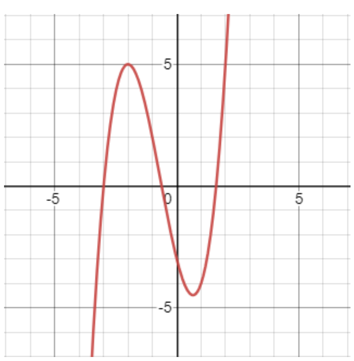
Answer:

(A) Try out numbers $x = 0, 1, -1, 2, -2, 3, -3, \ldots$ by plugging them into the expression $P(x) = x^3 + 2x^2 - 4x - 3 = 0$. It turns out that $f(-3) = 0$. By *Polynomial Remainder Theorem*, the polynomial $P(x)$ is divisible by $(x - (-3)) = x + 3$. Let us do a *long division* of polynomials:

$$\begin{array}{l}
x^3 + 2x^2 - 4x - 3 : x + 3 = x^2 - x - 1 \\
\underline{x^3 + 3x^2} \\
\quad -x^2 - 4x \\
\quad \underline{-x^2 - 3x} \\
\qquad -x - 3 \\
\qquad \underline{-x - 3} \\
\qquad\qquad 0
\end{array}$$

We have obtained that $P(x) = (x + 3)\left(x^2 - x - 1\right)$.

If you would like to guess roots for more complex polynomials, note that cubic functions can have up to 2 local maxima/minima and one inflection point – so guessing can be made easy by plotting a graph: https://bit.ly/3njQSb7.



(B) Apart from the root $x_1 = -3$ there is also the root where the other factor $x^2 - x - 1$ equals to 0. Let us solve this square equation. We get:

$$x_{2,3} = \frac{1 \pm \sqrt{1^2 + 4 \cdot 1 \cdot 1}}{2} = \frac{1 \pm \sqrt{5}}{2}.$$

(C) Due to the *Polynomial Remainder Theorem*, the original polynomial must also be divisible by $(x - x_2)$ and by $(x - x_3)$. After these divisions a number remains which must be 1 as the coefficient for x^3 in $P(x)$ is also 1.

Ultimately we get the following:

$$P(x) = (x+3)\left(x - \frac{1+\sqrt{5}}{2}\right)\left(x - \frac{1-\sqrt{5}}{2}\right).$$

Question 2: Consider the following truth table computing a Boolean function $f(x_1, x_2, x_3)$ of three variables:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
`false`	`false`	`false`	`true`
`false`	`false`	`true`	`false`
`false`	`true`	`false`	`false`
`false`	`true`	`true`	`false`
`true`	`false`	`false`	`false`
`true`	`false`	`true`	`true`
`true`	`true`	`false`	`true`
`true`	`true`	`true`	`true`

(A) Build a DNF given the truth table. Write all terms as conjunctions of exactly three variables or their negatons.

(B) Apply Boolean identities to create a shorter DNF.

(C) Write an expression for the Boolean function $f(x_1, x_2, x_3)$ using only implication and negation operations.

Answer:

(A) For the *full DNF* create one term per every line in the truth table, where $f(x_1, x_2, x_3) =$ `true`. If a variable on that line of the truth table must be `true`, write the variable as it is; if the variable must be `false`, precede that variable by a negation:

$$f = (\neg x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge x_3).$$

(B) Transform the expression with the last two terms like this:

$$(x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge x_3) \;\equiv\; (x_1 \wedge x_2) \wedge (\neg x_3 \vee x_3) \;\equiv\; (x_1 \wedge x_2).$$

Replacing this in the original DNF gives this Disjunctive Normal Form (which is not the full DNF):

$$f = (\neg x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee (x_1 \wedge x_2).$$

(C) There are many correct ways how to complete this task and get various (mutually equivalent) formulas containing the three variables, the implication $\rightarrow$ and negation $\neg$. Here is just one way – it observes that the truth table contains two subtables with 4 rows each which are truth tables for recognizable Boolean functions:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
`false`	`false`	`false`	`true`
`false`	`false`	`true`	`false`
`false`	`true`	`false`	`false`
`false`	`true`	`true`	`false`
`true`	`false`	`false`	`false`
`true`	`false`	`true`	`true`
`true`	`true`	`false`	`true`
`true`	`true`	`true`	`true`

Case1:
$\neg x_1 \wedge \neg(x_2 \vee x_3)$

Case2:
$x_1 \wedge (x_2 \vee x_3)$

$f(x_1, x_2, x_3) \equiv \mathit{Case1} \vee \mathit{Case2} \equiv \neg \mathit{Case1} \rightarrow \mathit{Case2}$

Case 1: If $x_1 = \texttt{false}$, then it is the negation of disjunction: $\neg(x_2 \vee x_3)$.

$$(\neg x_1) \wedge \neg(x_2 \vee x_2)) \;\equiv\; (\neg x_1 \wedge \neg(\neg x_2 \to x_3)) \;\equiv\; \neg(\neg x_1 \to \neg x_2 \to x_3).$$

Case 2: If $x_1 = \texttt{true}$, then this is the truth table of for disjuction: $(x_2 \vee x_3)$.

$$(x_1 \wedge (x_2 \vee x_2)) \;\equiv\; (x_1 \wedge (\neg x_2 \to x_3)) \;\equiv\; \neg(x_1 \to \neg(x_2 \to x_3)).$$

Now connect both cases with a disjunction:

$$(Case1) \vee (Case2) \equiv \neg(Case1) \to (Case2) \equiv \neg(\neg(\neg x_1 \to x_2 \to x_3)) \to \neg(x_1 \to \neg(x_2 \to x_3)).$$

Question 3: Consider the following argument: *If I gain weight, then I get depressed, and if I get depressed, then I eat too much. If I eat too much, then I get lazy, and if I get lazy, then I dont exercise. If I dont exercise, then I gain weight. Therefore, I will gain weight.*

(A) Define atomic propositions used in this argument (such propositions state simple facts, they do not contain if then constructs or negations). Denote every atomic proposition by a letter.

(B) Write all the sentences using the atomic propositions you defined in the previous step. Every sentence becomes a Boolean expression. Write these sentences into a single column – all of them except the last one are hypotheses. The last one (following the word Therefore is the conclusion).

(C) Is this argument valid? To check its validity, you could use e.g. the *Indirect truth table method* – try to make the conclusion false, and then see, which values the variables in the atomic propositions should be assigned certain values. If you can conclude that in order to make the conclusion false you need to make some hypothesis false as well, then the argument is valid. (On the other hand, if you can assign values so that all hypotheses are true, but the conclusion is still false, then the argument is not valid.)

Answer:

(A) List all the atomic statements (statements which cannot be further broken using Boolean operations):

- A – "I gain weight"
- B – "I get depressed"
- C – "I eat too much"
- D – "I get lazy"
- E – "I exercise"

One could also pick E – "I dont exercise", but this is not atomic, since it can be written as a negation of an even simpler statement.

(B) Now list all the premises and the conclusion using the atomic statements:

$$\begin{array}{l} A \to B \\ B \to C \\ C \to D \\ D \to \neg E \\ \neg E \to A \\ \hline \vdash\; A \end{array}$$

(C) The *Indirect truth table method* means a proof of validity (or invalidity) of some argument, which analyzes cases – but only focuses on the cases which are interesting: We only consider different ways how the conclusion could be false (still assuming that all the premises are correct). If we succeed, then the argument was invalid.

We start by assuming that $A = \texttt{False}$ (namely, I gain weight is not true even though all the premises are true). Let us build a (shortened) truth table showing the reasoning:

Step	A	B	C	D	E
Step 1	False				
Step 2	False				True
Step 3	False			False	True
Step 4	False		False	False	True
Step 5	False	False	False	False	True

Step 1: Assume that the conclusion is false.

Step 2: Assume that the premise $\neg E \rightarrow A$ is true, so $\neg E$ must be false (and, consequently, E must be true).

Step 3: Assume that the premise $D \rightarrow \neg E$ is true, so D must be false.

Step 4: Assume that the premise $C \rightarrow D$ is true, so C must be false.

Step 5: Assume that the premise $B \rightarrow C$ is true, so B must be false.

We now have the only possible assignment of truth values (in more complex cases the Indirect truth table method may lead to multiple subcases to be considered).

$$A = \texttt{False},\ B = \texttt{False},\ C = \texttt{False},\ D = \texttt{False},\ E = \texttt{True}.$$

Observe that this assignment of truth values makes all premises true, but the conclusion is false. Therefore the argument provided above is **invalid** (it is logically unsound and must never be used to conclude anything – no matter, whether the premises mentioned there are true or not).

Note: Any other proof method is also allowed. Exhaustive search over all variants how to assign truth values to 5 variables (A, B, C, D, E) leads to 32 subcases. Most of them are irrelevant, since it is just one line in the truth table out of 32 shows why the argument must be invalid.

Note: The above example is one case of *Circular reasoning* (see https://bit.ly/33KzIwh) – it shows how it is possible to reach wrong conclusions, if we already assume the conclusion and then move in a big circle of implications to the same conclusion once again.

Question 4:

(A) Prove that $\log_{12} 2$ is irrational.

(B) Does there exist a positive integer $a \in \mathbb{Z}_{>0}$ such that $\log_{12} a$ is rational fraction p/q that is not an integer? (Either prove that such a cannot exist or show a counterexample.)

Answer:

(A) Assume by contradiction that $\log_{12} 2 = \frac{p}{q}$ is a rational number and p/q is an irreducible fraction. In other words, we can assume that p and q are positive integers that are mutually prime (they do not have common divisors larger than 1).

Rewrite the equality:

$$\begin{aligned}
&\log_{12} 2 = \tfrac{p}{q}\\
&12^{\frac{p}{q}} = 2\\
&12^p = 2^q\\
&(2 \cdot 2 \cdot 3)^p = 2^q\\
&2^{2p} \cdot 3^p = 2^q
\end{aligned}$$

In the last equality the left-side expression $2^{2p} \cdot 3^p$ is divisible by 3 (whenever $p > 0$). The right-hand expression 2^q is not divisible by 3. Therefore they cannot be equal – a contradiction.

(B) Assume by contradiction that there exists an integer number a such that $\log_{12} a = \frac{p}{q}$ is a rational number and p/q is an irreducible fraction, which is **not** an integer.

Rewrite the equality:

$$\begin{aligned}
&\log_{12} a = \tfrac{p}{q} \\
&12^{\frac{p}{q}} = a \\
&12^p = a^q \\
&(2 \cdot 2 \cdot 3)^p = a^q \\
&2^{2p} \cdot 3^p = a^q
\end{aligned}$$

Express the number a as a product of prime numbers. Let 3^k (for some integer k) be the power of number 3 that is in this product of primes. In other words, the number a is divisible by 3^k, but not by 3^{k+1}. Express $a = N \cdot 3^k$, where N is the product of all the other prime numbers (except 3). For example, if $a = 216 = 2^3 \cdot 3^3$, then we must have $a = 8 \cdot 3^3$, i.e. $N = 8$ and $k = 3$. When we raise a to the power p, we must get $(N \cdot 3^k)^q = N^q \cdot 3^{kq}$.

Consider the equality obtained above – $2^{2p} \cdot 3^p = a^q$. The right side is divisible by 3^{kq} (the largest power of three), the left side is divisible by 3^p (also the largest power of three).

Since both sides of the equality must be the same, their largest powers of 3 must also be the same: $3^p = 3^{kq}$ or $p = kq$. We get that $p/q = k$. Therefore the fraction p/q must be an integer. This is a contradiction, since we asumed that $\log_{12} a$ is not an integer.

Question 5: Lidl supermarket chain sells identical cookies in two kinds of packages: package #1 contains 5 cookies, package #2 contains 13 cookies.

(A) Use the well-ordering principle to show that every amount of cookies $n \geq 48$ can be bought by selecting zero or more packages of either type.

(B) Show that 47 cookies cannot be bought in this manner.

Answer:

(A) For any natural $x \in \mathbb{N} = \{0, 1, 2, 3, \ldots\}$ define a predicate $P(x)$ which takes value `True` iff $x + 48$ cookies can be purhcased using some combination of the two available cookie packages (containing 5 and 13 cookies respectively).

Let us prove that this predicate is true for every natural x: $\forall x \in \mathbb{N}\ (P(x))$.

We know that $P(0) = P(1) = P(2) = P(3) = P(4) = \texttt{True}$. Indeed, all the cookie counts $(48, 49, 50, 51, 52)$ can be expressed using the packages available:

$$\begin{aligned}
48 &= 1 \cdot 13 + 7 \cdot 5 \\
49 &= 3 \cdot 13 + 2 \cdot 5 \\
50 &= 0 \cdot 13 + 10 \cdot 5 \\
51 &= 2 \cdot 13 + 5 \cdot 5 \\
52 &= 4 \cdot 13 + 0 \cdot 0
\end{aligned}$$

Now assume that there exists some integer $x \geq 5$ for which $P(x)$ is false. Due to the *well-ordering principle* there must exist the smallest x for which $P(x)$ is false. If $x \geq 5$ is the smallest value for which $P(x) = \texttt{False}$ then we must also have $P(x-5) = \texttt{False}$. Indeed, if you can buy $(x-5)+48$ cookies in Lidl, then you should also be able to buy $x + 48$ cookies – just take one more package containing 5 cookies.

So, the minimum element for which $P(x) = \texttt{False}$ cannot be $x \geq 5$. But it cannot take values $x = 0, 1, 2, 3, 4$ either (since we just expressed all numbers betwee 48 and 52 (see above). For this reason $P(x) = \texttt{False}$ is impossible for any $x \in \mathbb{N}$.

(B) One cannot purchase 47 cookies using just 13-cookie and 5-cookie packages. To see that you can consider 4 different cases – depending on whether you will use zero, one, two, or three packages containing 13 cookies:

- $47 - 0 \cdot 13 = 47$ is not divisible by 5.

- $47 - 1 \cdot 13 = 34$ is not divisible by 5.
- $47 - 2 \cdot 13 = 21$ is not divisible by 5.
- $47 - 3 \cdot 13 = 8$ is not divisible by 5.

So, one should not purchase $0, 1, 2$ or 3 packages containing 13 cookies (as the rest cannot be expressed as $5k$). But purchasing 4 packages containing 13 cookies is impossible too, since $47 < 4 \cdot 13 = 52$.

Question 6 (Supplementary Task):

Introduction: NAND gates are devices with two inputs x and y and one output z that compute the negation of conjunction: $z = \neg(x \wedge y)$ (not (x and y) also known as NAND operation). Denote NAND by $x \uparrow y$. It has the following truth table:

x	y	$z = x \uparrow y$
`false`	`false`	`true`
`false`	`true`	`true`
`true`	`false`	`true`
`true`	`true`	`false`

It is possible to express other Boolean operations using just the NAND ($\uparrow$) operation.

- Negation: $\neg x \equiv x \uparrow x$.
- Conjunction: $x \wedge y \equiv \neg(x \uparrow y) \equiv (x \uparrow y) \uparrow (x \uparrow y)$.
- Disjunction: $x \vee y \equiv \neg(\neg x \wedge \neg y) \equiv (\neg x \uparrow \neg y) \equiv ((x \uparrow x) \uparrow (y \uparrow y))$.
- Implication: $x \rightarrow y \equiv \neg x \vee y \equiv x \uparrow (y \uparrow y)$.

Problem: Assume that you need to use only NAND gates to compute XOR for two, three or four variables:

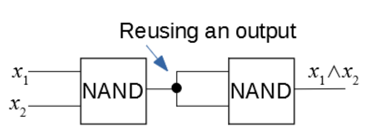
(A) $x_1 \oplus x_2$

(B) $x_1 \oplus x_2 \oplus x_3$

(C) $x_1 \oplus x_2 \oplus x_3 \oplus x_4$

Draw the circuits for these three expressions (A), (B), (C). If there are multiple solutions, select the circuit which uses as few NAND gates as possible. Justify your answers – why do the circuits compute the given expressions and why are they minimal.

Note: In a circuit you can reuse the output of the same NAND gate as input several times. The following image shows how $x_1 \wedge x_2$ can be computed using just 2 NAND gates. On the other hand, the formula $x_1 \wedge x_2 \equiv (x_1 \uparrow x_2) \uparrow (x_1 \uparrow x_2)$ contains three NAND operations, but there are two identical subexpressions $(x_1 \uparrow x_2)$.

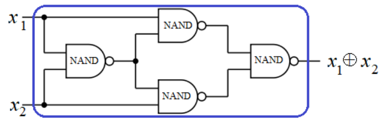


Answer:

We should verify the following Boolean equivalence:

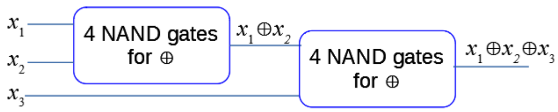
$$(x_1 \oplus x_2) \equiv \big((x_1 \uparrow x_2) \uparrow x_1\big) \uparrow \big((x_1 \uparrow x_2) \uparrow x_2\big)$$

(A) Using the above formula (and computing the subexpression $(x_1 \uparrow x_2)$ only once) gives us the following NAND-circuit for the expression $(x_1 \oplus x_2)$:



This expression uses 4 NAND gates.

(B) The longer expression $x_1 \oplus x_2 \oplus x_3$ can be obtained by first computing $x_1 \oplus x_2$ (using 4 gates), then XORing the result with x_3 (using 4 more gates):



The blue rounded rectanges contain copies of the NAND circuit for XOR – it is the same as in (A).

(C) Use 12 NAND gates to compute $(x_1 \oplus x_2) \oplus (x_3 \oplus x_4)$.