

Discrete Structures (W3): Quiz

Question 1. We have the following truth table. Find the correct DNF (Disjunctive Normal Form) expressing $E(X, Y, Z)$.

Note. Multiple answers may be true, since there may more than one way to write DNFs for the same Boolean function.

X	Y	Z	E
T	T	T	F
T	T	F	F
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

(A) $(X \wedge \neg Y \wedge Z) \vee (\neg X \wedge Y \wedge Z)$.

(B) $(\neg X \wedge Y \wedge Z) \vee (\neg Y \wedge Z)$.

(C) $(\neg X \wedge Z) \vee (X \wedge \neg Y \wedge Z)$.

(D) $(\neg X \wedge \neg Y \wedge Z) \vee (\neg X \wedge Y)$.

Question 2. Identify, which is the correct CNF (Conjunctive Normal Form) that computes the same thing as Peirce's arrow: $a \downarrow b = \neg(a \vee b)$.

Note. Multiple answers may be true, since there may more than one way to write CNFs for the same Boolean function.

(A) $(\neg a \vee \neg b) \wedge (\neg a \vee b) \wedge (a \vee \neg b)$.

(B) $(a \vee b) \wedge (\neg a \vee b) \wedge (a \vee \neg b)$.

(C) $\neg a \vee \neg b$.

(D) $\neg a \wedge (a \vee \neg b)$.

Question 3. We have predicates $A(p, i, r)$ (a Python program $p \in \mathcal{P}$ receives input $i \in \mathbb{Z}^+$ and answers with the result $r \in \mathbb{Z}^+$); $H(p, i)$ (program p halts on input i), and $C(i, r)$ (the correct/expected output for input i is r). Identify the predicate expression that expresses the following English sentence:

"For all inputs that are sufficiently large, the given program p halts and produces outputs that are correct; for some (finitely many) inputs p may loop forever, but it is never incorrect."

(A) $\forall p \in \mathcal{P} \forall i \in \mathbb{Z}^+ \forall r \in \mathbb{Z}^+, ((i > M \wedge H(p, i)) \vee (A(p, i, r) \wedge C(i, r)))$.

(B) $\exists M \in \mathbb{Z}^+ \forall i \in \mathbb{Z}^+ \exists r \in \mathbb{Z}^+, ((i < M \wedge \neg H(p, i)) \vee (A(p, i, r) \wedge C(i, r)))$.

(C) $\forall p \in \mathcal{P} \exists M \in \mathbb{Z}^+ \forall i \in \mathbb{Z}^+ \forall r \in \mathbb{Z}^+, ((i > M \wedge H(p, i)) \vee (A(p, i, r) \wedge C(i, r)))$.

(D) $\forall p \in \mathcal{P} \forall i \in \mathbb{Z}^+ \forall r \in \mathbb{Z}^+, ((i > M \wedge H(p, i)) \vee (A(p, i, r) \wedge C(i, r)))$.

Question 4. For the quantifiers $(A(p, i, r), H(p, i), C(i, r))$ defined in the previous question, the following expression is given:

$$\forall p \in \mathcal{P} \exists i \in \mathbb{Z}^+ \exists j \in \mathbb{Z}^+,$$

$$(j > i \wedge (H(p, i) \rightarrow H(p, j))).$$

Which sentence is expressed by this?

- (A) Each Python program halts for at least 2 inputs.
 (B) Each Python program halts for infinitely many inputs.
 (C) There is no Python program that halts on exactly 1 input.
 (D) For each Python program there is the largest input for which it halts.
 (E) The statement is always true (regardless of halting behavior of Python programs).

Question 5. There is a set of 2 students $S = \{s_1, s_2\}$ and a set of 5 chairs $C = \{c_1, c_2, c_3, c_4, c_5\}$. Find, how many such functions $f : S \rightarrow C$ exist (a function by definition is a mapping that assigns a chair for each student; it is NOT always true that different students get different chairs or that all chairs are occupied). Please find the total number of such functions, also the number of injective, surjective and bijective functions among them.

Note. Your answer should be a comma-separated list of 4 numbers (and 3 commas).

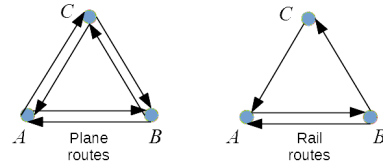
Question 6. Predicate $S(x, y, z)$ is true iff $x \cdot y = z$. Given the following equation:

$$\forall u \in \mathbb{Z}^+ \forall v \in \mathbb{Z}^+ \forall d \in \mathbb{Z}^+ \forall p \in \mathbb{Z}^+ \forall q \in \mathbb{Z}^+ \exists k \in \mathbb{Z}^+, S(x, u, y) \wedge S(x, v, z) \wedge S(d, p, y) \wedge S(d, q, z) \rightarrow S(d, k, x).$$

Which statement is expressed by this expression?

- (A) d is the greatest common divisor of y and z ,
 (B) x is the greatest common divisor of p and q ,
 (C) x is the greatest common divisor of y and z ,
 (D) d is the greatest common divisor of u and v .

Question 7.



Find the predicate expression that expresses this statement: "For the two given cities x and y there is a two-leg trip using planes, but there is no two-leg trip using rail."

Note. We call a trip "two-leg", if it uses exactly two plane flights or exactly two rail links. In this example all variables are from a 3-element set $\text{City} = \{A, B, C\}$ – see Figure.

(A) $\forall x \forall y \forall z, \text{Plane}(x, z) \wedge \text{Plane}(z, y) \wedge \neg(\text{Rail}(x, z) \wedge \text{Rail}(z, y))$.

(B) $\exists z, \text{Plane}(x, z) \wedge \text{Plane}(z, y) \wedge \neg(\text{Rail}(x, z) \wedge \text{Rail}(z, y))$.

(C) $\forall x \forall y \exists z, \text{Plane}(x, z) \wedge \text{Plane}(z, x) \wedge (\neg \text{Rail}(x, z) \vee \neg \text{Rail}(z, y))$.

(D) $\exists z_1 \forall z_2, \text{Plane}(x, z_1) \wedge \text{Plane}(z_1, y) \wedge (\neg \text{Rail}(x, z_2) \vee \neg \text{Rail}(z_2, y))$.

Answers

Question 1. Answer (B),(C).

An obvious way to write DNF for this expression:

$$(X \wedge \neg Y \wedge Z) \vee (\neg X \wedge Y \wedge Z) \vee (\neg X \wedge \neg Y \wedge Z).$$

It contains one clause per every T in the truth table. But it is not among the answers. Instead the answers (B) and (C) contain one shorter clause (with just two variables) that is equivalent to two clauses with three variables. For example,

$$\begin{aligned} (\neg Y \wedge Z) &\equiv \\ \equiv \text{True} \wedge (\neg Y \wedge Z) &\equiv \\ \equiv (X \vee \neg X) \wedge (\neg Y \wedge Z) &\equiv \\ \equiv (X \wedge (\neg Y \wedge Z)) \vee (\neg X \wedge (\neg Y \wedge Z)) \end{aligned}$$

Therefore the DNF can be written in a more compact form when we combine two blue clauses into a shorter one:

$$\begin{aligned} (X \wedge \neg Y \wedge Z) \vee (\neg X \wedge Y \wedge Z) \vee (\neg X \wedge \neg Y \wedge Z) &\equiv \\ \equiv (\neg Y \wedge Z) \vee (\neg X \wedge Y \wedge Z) \end{aligned}$$

Question 2. Answer: (A), (D).

The variant (A) lists all 3 ways how $a \downarrow b$ can take value F.

Variant (D) $(\neg a) \wedge (a \vee \neg b)$ is the same thing. The clause $(\neg a)$ there is equivalent to two clauses from (A): $(\neg a \vee \neg b) \wedge (\neg a \vee b)$.

Question 3. Answer: (B).

This means that there is a positive integer M such that for each $i \in \mathbb{Z}^+$ there would be $r \in \mathbb{Z}^+$ such that **either** input i is small ($i < M$) and program does not halt. **Or** program p outputs result r upon receiving i and that result is correct. (Note that even for small i it is allowed to stop and to output a correct answer: In this case $(i < M \wedge \neg H(p, i))$ is false, but $(A(p, i, r) \wedge C(i, r))$ is true.

Let us consider answer alternative (C):

$$\begin{aligned} \forall p \in \mathcal{P} \exists M \in \mathbb{Z}^+ \forall i \in \mathbb{Z}^+ \forall r \in \mathbb{Z}^+, \\ ((i > M \wedge H(p, i)) \vee (A(p, i, r) \wedge C(i, r))). \end{aligned}$$

Note that for the answer (C) (just like (A) and (D)) starts with $\forall p \in \mathcal{P}$. These statements contain p as a *bound variable*; therefore they cannot express anything meaningful about the special p mentioned in the English sentence: “For all inputs that are sufficiently large, **the given program** p halts and produces outputs.” (The given Python program p is not **any program** or **some program**; it is one certain program – variable p should have no quantifier.)

Even if we drop that $p \in \mathcal{P}$ the statement (C) expresses something different: “For a program there exists a number M such that for all $i > M$ the program should halt. For any “small” input i ($i \leq M$) it is not sufficient to halt; the program must be able to output **any** result r for the received input; and moreover – that output should be correct: $(A(p, i, r) \wedge C(i, r))$).

Question 4. Answer: (E).

Let us forget anything we know about Python programs and their halting. Let $H(p, i)$ be any predicate on elements of some set $p \in \mathcal{P}$ and positive integers i . We can prove that $(j > i \wedge (H(p, i) \rightarrow H(p, j)))$ is always true. For the given p take three different numbers i_1, i_2, i_3 . Among $H(p, i_1), H(p, i_2), H(p, i_3)$ there are two values that are both true (or both false). Assign i to the smallest of these i_1, i_2, i_3 , and assign j to the largest.

Question 5. Answer: 25, 20, 0, 0.

There are $5 \cdot 5 = 25$ ways to assign two students to five chairs (you have to make a choice “1 of 5” two times). Out of these 25 functions there are 5 functions that map both students to the same chair (c_1, c_2, c_3, c_4 , or c_5). All the other $25 - 5 = 20$ functions that are injective (i.e. $f(s_1) \neq f(s_2)$).

Since there are more chairs than students, there are no surjective and bijective functions.

Question 6. Answer: (C).

The only option can be x is the greatest common divisor of y and z . $S(x, u, y)$ and $S(x, v, z)$ show that x is **some** common divisor of y and z . And since $S(d, p, y) \wedge S(d, q, z) \rightarrow S(d, k, x)$ it means that another common divisor d is also a divisor of x .

Therefore x is the **greatest** common divisor of y and z .

You can easily eliminate alternatives (B), (D), because they state something about greatest common divisor of p, q or u, v . But all these variables are **bound** variables (they do not have any specific value; they vary over the whole set of positive integers).

Question 7. Answer: (D).

Note that the intermediate city z_1 for plane flights (that should exist) has nothing common with the city z_2 for rail connections (that should not exist). The conjunction **BUT** means same as **AND** from the logic point of view. We could express the answer like this:

$$\begin{aligned} (\exists z_1, \text{Plane}(x, z_1) \wedge \text{Plane}(z_1, y)) \text{ AND} \\ \neg(\exists z_2, \text{Rail}(x, z_2) \wedge \text{Rail}(z_2, y)). \end{aligned}$$

We can simplify the second clause using De Morgan’s law: $\forall z_2, (\neg \text{Rail}(x, z_2) \vee \neg \text{Rail}(z_2, y))$ and combine with the first clause by writing both quantifiers $\exists z_1$ and $\forall z_2$ at the very front of the expression.