Discrete Sample Quiz 3

Question 1. *Definition.* A Boolean formula is a *Conjunctive Normal Form (CNF)*, if it is a conjunction of one or more clauses, where a clause is a disjunction of literals. Each literal is either a variable ($u, v, \ldots$) or its negation ($\neg u, \neg v, \ldots$).

Find the CNF computing the following truth table for Boolean expression $E(a, b, c)$:

a	b	c	$E(a,b,c)$
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	T

Question 2. The following CNF is given:

$$E = (a \vee \neg c) \wedge (b \vee \neg b).$$

Find 2 Boolean expressions equivalent to E:

(A) $a \rightarrow \neg b$,

(B) $a \rightarrow c$,

(C) $\neg c \rightarrow \neg a$,

(D) $(a \wedge c) \rightarrow b$,

(E) $c \rightarrow a$,

(F) $a \rightarrow (c \rightarrow b)$,

(G) $\neg a \rightarrow \neg c$.

Question 3. We have six statements about Python programs $p \in \mathcal{P}$ that convert inputs $i \in \mathbb{Z}^+$ into results $r \in \mathbb{Z}^+$. A Python program may either loop indefinitely or halt (i.e. it eventually stops).

1. Some programs return the correct result for all possible inputs and they never loop indefinitely.
2. For any program one can find another program such that it returns the same result for the same inputs as the first one (or loops indefinitely, **iff** the first program does the same).
3. There is a program that only loops indefinitely for at most finitely many inputs (or maybe none at all), but for all other inputs it produces the correct result.
4. There is at least one Python program that always halts, and for sufficiently large inputs it produces the correct result, but it may err for some small-size inputs.
5. For a program to produce a correct result for some input i it is strictly necessary to halt.
6. A Python program always produces exactly one result for the given input provided that it halts.

We use these 3 predicates:
$A(p_1, i_2, r_3)$ is true iff Python program $p_1 \in \mathcal{P}$ receives input $i_2 \in \mathbb{Z}^+$ and outputs result $r_3 \in \mathbb{Z}^+$.
$H(p_1, i_2)$ is true iff program $p_1 \in \mathcal{P}$ receives input i_2 and halts (i.e. does not loop indefinitely).
$C(i_1, r_2)$ is true iff for input i_1 the correct result is r_2.

Note. Write your answer as a comma-separated list: For example, `F,E,D,C,B,A` tells that the 1st statement is **(F)**, the 2nd one is **(E)**, ..., the last one is **(A)**.

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

(B) $\forall p_1 \in \mathcal{P}\ \exists p_2 \in \mathcal{P}\ \forall i \in \mathbb{Z}^+\ \forall r \in \mathbb{Z}^+,$
$((\neg H(p_1,i) \wedge \neg H(p_2,i)) \vee (A(p_1,i,r) \leftrightarrow A(p_2,i,r)))$

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

(D) $\forall p \in \mathcal{P}\ \forall i \in \mathbb{Z}^+\ \forall r_1 \in \mathbb{Z}^+\ \forall r_2 \in \mathbb{Z}^+,$
$(H(p,i) \wedge A(p,i,r_1) \wedge A(p,i,r_2) \ \rightarrow\ r_1 = r_2).$

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

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

Question 4. There is a set of 4 students $S = \{s_1, s_2, s_3, s_4\}$ and a set of 2 chairs $C = \{c_1, c_2\}$. Find, how many such functions $f : S \rightarrow C$ exist, how many of them are injective, surjective and bijective.
Note. Your answer should be a comma-separated list of 4 numbers (and 3 commas): `total,injective,surjective,bijective`.

Question 5. There is a predicate $S(x, y, z)$ defined for triplets of positive integers, $S : \mathbb{Z}^+ \times \mathbb{Z}^+ \times \mathbb{Z}^+ \rightarrow \{\mathtt{T}, \mathtt{F}\}$. $S(x, y, z)$ is true iff $x \cdot y = z$.
Express these statements about positive integers using only $S(x, y, z)$, Boolean operations and quantifiers.
Note. Please use only the predicate $S(\ldots)$ in your answers, avoid any other predicates or relations (such as equality, divisibility, etc.).

(A) $x/y = z$,

(B) $x = 1$,

(C) $x = y$,

(D) x is divisible by y (i.e. $y \mid x$).

(E) x has odd number of positive divisors.

(F) x is not a prime.

(G) x is a prime.

Answers

Question 1. Answer:

$$(\neg a \vee \neg b \vee c) \wedge (\neg a \vee b \vee \neg c) \wedge (\neg a \vee b \vee c) \wedge$$
$$\wedge (a \vee \neg b \vee c) \wedge (a \vee b \vee \neg c)$$

Every value F in the truth table produces one clause in this expression. For example $E(\mathtt{T},\mathtt{T},\mathtt{F}) = \mathtt{F}$ is addressed by a disjunction $(\neg a \vee \neg b \vee c)$.

If we combine red and blue clauses, we can rewrite this into a more compact expression, which is also CNF. Note, that this is not the only way to rewrite:

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

Question 2. Answer: **(E)**, **(G)**.
$(a \vee \neg c) \wedge (b \vee \neg b) = (a \vee \neg c) \wedge \mathtt{T} = (a \vee \neg c)$. This is same as $c \rightarrow a$ or the contrapositive variant of the same implication: $\neg a \rightarrow \neg c$.

Question 3. Answer:

1	2	3	4	5	6
(F)	**(B)**	**(C)**	**(E)**	**(A)**	**(D)**

In **(B)** we should write: the second program loops indefinitely **iff** the first program does so. If we write **if** instead, it would mean only one-way conditional, which does not mean equivalence between the programs p_1 and p_2, but something more complicated.

In **(B)** you could write $(H(p_1,i) \leftrightarrow H(p_2,i))$ instead of $(\neg H(p_1,i) \wedge \neg H(p_2,i))$ or even omit any reference to $H(\ldots)$ predicates. The notation $(H(p_1,i) \leftrightarrow H(p_2,i))$ stresses the fact that p_1 and p_2 behave in the same way. But in fact the equivalence $(A(p_1,i,r) \leftrightarrow A(p_2,i,r))$ implicitly states the fact that p_1 and p_2 are defined on the same inputs.

Please note that **(B)** should **NOT** have quantifier $\exists r \in \mathbb{Z}^+$ (instead of $\forall r \in \mathbb{Z}^+$), because we need equivalence for all positive integers r. If we only care about the actual result of p on input i, then it is possible to "cheat" the logic formula: pick result r which is impossible for both p_1 and p_2; then $A(p_1,i,r)$ and $A(p_2,i,r)$ would both be false (and thus equivalent).

In **(C)** one cannot replace $\exists r \in \mathbb{Z}^+$ with $\forall r \in \mathbb{Z}^+$ (as in an earlier draft of this test), because it does not make sense to ask that a program p on input i produces **each** positive integer result; and all results turn out to be correct. (Such a formula may have some useful meaning if predicates A, H, C have different interpretations, but for Python programs this is technically correct, but has an absurd meaning: that a given program produces any result whatsoever.)

In **(E)** it is absolutely necessary to write a quantifier $\forall r \in \mathbb{Z}^+$, because leaving r as a free variable (without a quantifier) would mean that the Python program always produces the same result (the value of r). It is not

Question 4. Answer: `16,0,14,0`
Please note that there can be no injections (and therefore also bijections): students will always collide – run to the same chair, since there are fewer chairs than students.

The total number of functions is $2^4 = 16$. Each student can choose between 2 chairs, so it is $2 \times 2 \times 2 \times 2 = 16$. Moreover, there are only 2 "constant" functions (where all four students go to c_1 or to c_2). They are not surjective, because some chairs stay unoccupied in this case. All the other $16 - 2 = 14$ functions from S (4 elements) to C (2 elements) are surjections.

Question 5. Here we write all 7 statements with predicate $S(x,y,z)$ and quantifiers.

(A) $S(z,y,x)$ (or, equivalently, $S(y,z,x)$).

(B) $\exists y \in \mathbb{Z}^+,\ S(x,y,y)$.
Use the property: $x = 1$ can multiply with an y, and does not increase/decrease.
You could also write simply $S(x,x,x)$. Indeed, x is the only positive integer such that $x \cdot x = x$. (Allowing $x = 0$ would spoil this, because $0^2 = 0$.)

(C) $\exists z \in \mathbb{Z}^+,\ S(x,z,y) \wedge S(z,z,z)$.
So, $x = y$ iff there is a $z = 1$ such that $x \cdot z = y$.

(D) $\exists d \in \mathbb{Z}^+,\ S(y,d,x)$.

(E) $\exists y \in \mathbb{Z}^+,\ S(y,y,x)$.
This means that $y^2 = x$ for some y; hence x is a full square and should have odd number of divisors.

(F) $\exists y \in \mathbb{Z}^+\ \exists z \in \mathbb{Z}^+,\ (S(y,z,x) \wedge \neg S(y,y,y) \wedge \neg S(z,z,z))$.
This means that a non-prime x can be expressed as a product of two positive integers y, z, where neither of them is 1.)

(G) $\forall y \in \mathbb{Z}^+\ \forall z \in \mathbb{Z}^+,\ (\neg S(y,z,x) \vee S(y,y,y) \vee S(z,z,z))$.
We apply negation to the previous quantifier formula; all $\exists$ quantifiers turn into $\forall$; formulas are changed according to De Morgan's laws.