Question 1. Define the predicates

$$P(n)\colon \text{ For any set } A, \text{ if } |A| = n \text{ then } |\mathcal{P}(A)| = 2^n$$

$$Q(A, n)\colon |A| = n \Longrightarrow |\mathcal{P}(A)| = 2^n$$

a) Prove $\forall n \in \mathbb{N}, P(n)$.

Proof. **Base Case.** To show $P(0)$, consider any set A such that $|A| = 0$. Then $A = \varnothing$ and its only subset is $\varnothing$. Thus $\mathcal{P}(A) = \{\varnothing\} \implies |\mathcal{P}(A)| = 1 = 2^0$, verifying that $P(0)$ is true.

Induction Step. Suppose $P(k)$ holds for some $k \in \mathbb{N}$. Now $P(k+1)$ will be proven to hold. Let A be a set such that $|A| = k+1$. $k+1$ is at least 1, so A possesses at least one element, which will be denoted as a.

Consider the set $A \setminus \{a\}$. Since $\big|A \setminus \{a\}\big| = k$, by the induction hypothesis,

$$\big|\mathcal{P}(A \setminus \{a\})\big| = 2^k$$

Notice that $\mathcal{P}(A \setminus \{a\})$ contains all the subsets of A that do not contain a. The remaining subsets must all contain a. The remaining subsets of A can be obtained by taking every individual element in $\mathcal{P}(A \setminus \{a\})$ and unioning it with $\{a\}$. Thus A contains twice as many subsets as $A \setminus \{a\}$. In mathematical terms,

$$|\mathcal{P}(A)| = 2 \cdot |\mathcal{P}(A \setminus \{a\})| = 2 \cdot 2^k = 2^{k+1}$$

It has been shown that $P(k+1)$ holds.

By the principle of simple induction, $\forall n \in \mathbb{N}, P(n)$.

□

b) Prove that for every set A, $\forall n \in \mathbb{N}$, $Q(n)$. This method does not work. Here is the attempt at the proof:

Proof. Fix a set A. Proceed with using simple induction.

Base Case. Let $n = 0$. To show $Q(A, n)$ holds, suppose that $|A| = 0$. Then $A = \varnothing$. Thus $\mathcal{P}(A) = \{\varnothing\} \implies |\mathcal{P}(A)| = 1 = 2^0$.

Thus $Q(A, 0)$.

Induction Step. Suppose that $Q(A, k)$ holds for some $k \in \mathbb{N}$. To show $Q(A, k+1)$, suppose $|A| = k+1$. However, this is where the problem arises.

The induction hypothesis cannot be utilised since our assumption requires $|A| = k+1$, while the condition to use the induction hypothesis is $|A| = k$.

Thus the proof by induction cannot be continued.

□

Question 2. Let $n, m \in \mathbb{N}$. Let A, B be arbitrary finite sets of size m and n respectively.

a) How many fuctions are there with domain A and co-domain B? It can be shown using simple induction on m that the answer to this question is n^m.

Proof. First, particular edge cases will be examined. For $n, m \in \mathbb{N}$, define the predicate

$$P(m)\colon \text{ For every positive natural } n \text{, there are } n^m \text{ functions with finite domain of size } m$$
$$\text{and finite co-domain of size } n$$

Fix $m \in \mathbb{N}$.

Base Case. Let $m = 0$. There are no functions that can map to nothing, therefore the number of functions is $0^n = 0$.

Induction Step. Suppose that $P(n, k)$ holds for every $n \in \mathbb{N}$, but only for some $k \in \mathbb{N}$. Let A, B be finite sets such that $|A| = k + 1$ and $|B| = n$. A contains at least one element a. Consider the set $A \setminus \{a\}$. By the induction hypothesis, there are n^k functions with domain $A \setminus \{a\}$ and co-domain B. For every such function f_k and some $b \in B$, define a new function

$$f_b(x) = \begin{cases} f_k(x), & \text{if } x \in A \setminus \{a\}; \\ b, & \text{if } x = a; \end{cases}$$

Every function that maps elements from A to B can be written in this form. Thus there are $n^k * n = n^{k+1}$ fuctions that map from A to B.

□

b) Use part (a) to prove the original statement in Q1 directly without the use of induction.

Proof. Let A be a set such that $|A| = n$. Every subset A' of A can be defined as a function f that maps elements of A to $\{0, 1\}$:

$$\text{For any } a \in A, \text{ if } a \in A', f(a) = 1. \text{ Otherwise, } f(a) = 0$$

From the previous part, there are 2^n different functions with domain A and co-domain $\{0, 1\}$, which also means that there are 2^n subsets of A, which implies that $\mathcal{P}(A) = 2^n$.

□

Question 3. In propositional logic, you have seen the connectives $\neg, \wedge, \vee, \rightarrow$, and $\leftrightarrow$. Prove using structural induction that any proposition built using these connectives is equivalent to a proposition built only using $\neg, \rightarrow$.

Proof. The proof will be done using structural induction.

For a proposition P, Define the predicate

$$Q(P) : P \text{ is equivalent to some proposition built only using } \neg,\ \rightarrow$$

Base Case. For all $P_i(x_{j_1}, x_{j_2}, \ldots, x_{j_k})$, they are equivalent to a proposition built only using $\neg, \rightarrow$, which are themselves. Thus $Q(P_i(x_{j_1}, x_{j_2}, \ldots, x_{j_k}))$ holds.

Induction Step. Let A, B be propositions such that $Q(A)$ and $Q(B)$ hold. It follows that $A \equiv C, B \equiv D$, where C, D are propositions built from only $\neg,\ \rightarrow$. 5 recursive cases will be considered.

1. $\neg A \equiv \neg C$, thus $Q(\neg A)$ holds
2. $A \implies B \equiv C \implies D$, $Q(A \implies B)$ holds
3. $A \wedge B \equiv C \wedge D$. It will be shown using a truth table that $C \wedge D \equiv \neg(C \implies \neg D)$.

C	D	$C \wedge D$	$\neg(C \implies \neg D)$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

Since $A \wedge B \equiv \neg(C \implies \neg D)$, which is a proposition built from only $\neg, \implies$. Thus $Q(A \wedge B)$ holds.

4. $A \vee B \equiv C \vee D$. It will be shown using a truth table that $C \vee D \equiv \neg(\neg C \implies D)$.

C	D	$C \vee D$	$\neg(\neg C \implies D)$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

Since $A \vee B \equiv \neg(\neg C \implies D)$, which is a proposition built from only $\neg, \implies$. Thus $Q(A \vee B)$ holds.

5. $A \iff B \equiv C \iff D$. It will be shown using a truth table that $C \iff D \equiv \neg((C \implies D) \implies \neg(D \implies C))$.

C	D	$C \iff D$	$\neg((C \implies D) \implies \neg(D \implies C))$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	T	T

Since $C \iff D \equiv \neg((C \implies D) \implies \neg(D \implies C))$, which is a proposition built from only $\neg, \implies$. Thus $Q(A \iff B)$ holds.

By the principle of structural induction, $Q(P)$ holds for all propositions P.

$\square$

Question 4. Consider the following two-pointer style Python program which finds whether a given string s is a palindrome or not:

```
def check_if_palindrome(s):
    left = 0
    right = len(s) - 1
    while left < right:
        if s[left] != s[right]:
            return False
        left += 1
        right -= 1
    return True
```

Prove correctness and termination. Clearly state the loop variant and the loopinvariant, and use induction properly.

Proof. □