Question 1: Hair Splitting with Set Expressions

Let us define the successor of the set $A$ to be the set $A \cup \{A\}$. Find the successors of the following sets:

a) $A = \{x\}$
   $$A \cup \{A\} = \{x, \{x\}\}$$

b) $B = \{x, y\}$
   $$B \cup \{B\} = \{x, y, \{x, y\}\}$$

c) $C = \emptyset$
   $$C \cup \{C\} = \{\emptyset\}$$

d) $D = \{\emptyset, \{\emptyset\}\}$
   $$D \cup \{D\} = \{\emptyset, \{\emptyset\}, \emptyset, \{\emptyset\}\}$$

Question 2: Tautologies and Contradictions

Find out for each of the following propositions whether it is a tautology, a contradiction, or neither (a contingency). Prove your answer.

a) $[(p \rightarrow q) \land (q \rightarrow r)] \rightarrow (p \rightarrow r)$

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \rightarrow q</th>
<th>q \rightarrow r</th>
<th>(p \rightarrow q) \land (q \rightarrow r)</th>
<th>(p \rightarrow r)</th>
<th>[(p \rightarrow q) \land (q \rightarrow r)] \rightarrow (p \rightarrow r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

It is a tautology!
b) \((p \lor q \lor r) \rightarrow [(q \rightarrow r) \lor (p \rightarrow q)]\)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Another tautology!

**Question 3: Set Operations**

Let us take a look at the sets \(A = \{x, y, z\}, B = \{1, 2\}, C = \{y, z\}\). List the elements of the following sets \(D, E, F, G, H,\) and \(I\):

a) \(D = (B \times A) - (B \times C) = \{(1, x), (2, x)\}\)

b) \(E = 2^A - 2^C = \{\{x\}, \{x, y\}, \{x, z\}, \{x, y, z\}\}\)

c) \(F = 2^{2^B} = 2^{\{\emptyset, \{1\}, \{2\}, \{1, 2\}\}} = \{\emptyset, \emptyset, \{\emptyset\}, \{\{1\}\}, \{\{2\}\}, \{1\}, \{2\}, \{\{1\}, \{2\}\}, \{\{1\}, \{1\}\}, \{\{2\}, \{2\}\}, \{\{1\}, \{2\}\}, \{\{1\}, \{1\}\}, \{\{2\}, \{2\}\}, \{\{1\}, \{1\}\}, \{\{2\}, \{2\}\}, \{\{1\}, \{1\}\}, \{\{2\}, \{2\}\}, \{\{1\}, \{1\}\}, \{\{2\}, \{2\}\}, \{\emptyset, \emptyset, \emptyset, \emptyset\}\}\}

d) \(G = (A \times B \times C) \cap (C \times B \times A) = \{(y, 1, y), (y, 1, z), (z, 1, y), (z, 1, z), (y, 2, y), (y, 2, z), (z, 2, y), (z, 2, z)\}\)

e) \(H = \{(a, b, c) \mid a, b, c \in B \land b \neq c \land a = b\} = \{(1, 1, 2), (2, 2, 1)\}\)

f) \(I = \{(a, b, c) \mid a \in A \land b \in B \land c \in C \land a = c\} = \{(y, 1, y), (z, 1, z), (y, 2, y), (z, 2, z)\}\)
Question 4: Cardinality

Are the following statements true for all sets A, B and C? Prove your answers.

a) \(|A \cup B \cup C| = |A - B - C|

No. Counterexample: A = \{1\}, B = \{1\}, C = \{1\} gives us 1 = 0

b) \(|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C|

No. Counterexample: A = \{1\}, B = \{1\}, C = \{1\} gives us 1 = 0

Question 5: Functions

Find out whether the following functions from \(\mathbb{R}\) to \(\mathbb{R}\) are injective, surjective, and/or bijective (no proof necessary).

a) \(f(z) = -z\)

Injective, surjective, and bijective.

b) \(f(z) = 300z^5 + 4\)

Injective, surjective, and bijective!

c) \(f(z) = z \cdot \sin z\)

Not injective: For example, \(f(0) = f(\pi) = 0\)

Surjective, because with growing \(z\) we can reach any value for \(f(z)\).

Not bijective.

d) \(f(z) = z^2/(z^2 + 1)\)

Not injective. For example, \(f(1) = f(-1) = 0.5\)

Not surjective. For example, there is no \(z\) such that \(f(z) = -1\)

Question 6: Big-O Estimates

Give as good a big-O estimate as possible for the following complexity functions:
a) \( f(n) = (n \cdot \log n) (n^2 + 2n) = n^3 \cdot \log n + 2n^2 \cdot \log n \)

Term with greater exponent will dominate, therefore it is \( O(n^3 \cdot \log n) \).

b) \( f(n) = (2n! + 4n^3) + (2^n \cdot n^3) \)

Factorial will dominate, so it is \( O(n!) \).

c) \( f(n) = n^4 + 5 \log n + n^3 (n^2 + 2n) = n^4 + 5 \log n + n^5 + 2n^4 \)

Again, the greatest exponent decides: \( O(n^5) \).

**Question 7: Algorithms and Their Complexity**

a) Write a simple program in pseudocode (or in Python, C, C++, or Java, but only use basic commands so that comparisons can be counted) that receives a sequence of integers \( a_1, \ldots, a_n \) as its input and determines if the sequence contains two distinct terms \( x, y \) such that \( x = y^2 \). Once it finds such terms, it prints them and terminates; it does not continue searching after the first find. If the program does not find any such terms, it prints a disappointed comment and also terminates.

Example program with test in plain C:

```c
#include <stdio.h>

void findSquarePair(int seq[], int length)
{
    int x, y;
    for (x = 0; x < length; x++)
        for (y = 0; y < length; y++)
            if (seq[x] != seq[y] && seq[x] == seq[y]\*seq[y])
                {
                    printf("Awesome! I found %d and %d!\n", seq[x], seq[y]);
                    return;
                }  
    printf("How frustrating! I didn't find any such pair!\n");
}

int main()
{
    int testSeq[10] = { 8, -3, 4, -2, 10, 9, 7, 20, -5, 7 };
    findSquarePair(testSeq, 10);
    return 0;
}
```

b) Describe the kind of input that causes worst-case time complexity for your algorithm (only count comparisons), and explain why this is the case.
The most comparisons for a given sequence length are necessary if there is no $x = y^2$ match among its elements at all and the sequence does not contain any duplicates. In that case, in each iteration of the loop two comparisons (plus the one in the “for” construct) have to be performed.

c) Provide an equation for your algorithm that describes the number of required comparisons as a function of input length $n$ in the worst case. For some algorithms, it may be a good idea to first use a sum notation, but at the end you should provide a closed-form equation, i.e., one that no longer uses the sum symbol but only operations such as multiplication or addition of individual numbers or variables.

If you count the comparisons in the “for” constructs (and if you did not, you will not lose points), then for a sequence of $n$ elements, the inner loop requires $(3n + 1)$ comparisons (the “+1” occurs at the end when deciding to leave the loop). The outer loop has one comparison in the “for” construct, loops $n$ times and needs one more comparison for deciding that the search failed. So the total number of comparisons $f(n)$ is:

$$f(n) = n \cdot (3n + 1) + n + 1 = 3n^2 + 2n + 1$$

d) Use the big-O-notation to describe the worst-case time complexity of your algorithm.

$O(n^2)$

**Question 8 (Bonus Question): Venn Diagrams**

Draw the Venn diagrams for the following sets:

a) $A \cup B \cup C$
b) \( A \cup (B - C) \)

\[
\begin{array}{c}
A \\
\cap \\
B \\
\cap \\
C
\end{array}
\]

\( A \cap B \cup (A \cap C) \)

\[
\begin{array}{c}
A \\
\cap \\
B \\
\cap \\
C
\end{array}
\]
d) \((A \cap B) \cup (A \cap \overline{C})\)