Solutions of final review problems

1. Verify with the help of diagrams, which of the following statements hold in general. In order to make your conclusions draw the diagrams of the left hand side and the right hand side of these expressions.

(i) \( A-(B\cup C)=(A-B)-C \)
(ii) \( (A-B)\cap C \subseteq (A\cup C)-B \)

Solution:
(i) Both left and right hand sides are represented by the following diagram

The equality holds.
(ii)
LHS
2. Is it true that for all finite sets $A, B, C$

\[\#((A \cap B) \cup (A \cap C) \cup (B \cap C)) = \]
\[\#(A \cap B \cap C) + \#(\sim A \cap B \cap C) + \#(A \cap \sim B \cap C) + \#(A \cap B \cap \sim C)?\]

Solution:
We observe that sets
\[A \cap B \cap C, \sim A \cap B \cap C, A \cap \sim B \cap C, A \cap B \cap \sim C\]
are pairwise disjoint and that their union equals
\[(A \cap B) \cup (A \cap C) \cup (B \cap C).\] We conclude that the equality holds.

3. For each of the following laws of set theory find a corresponding law of propositional logic
(i) $(A \subseteq B \land B \subseteq C) \Rightarrow (A \subseteq C)$
(ii) $A \subseteq B \equiv \sim B \subseteq \sim A$

Solution:
Let $A = \{x \in U \mid p\}, B = \{x \in U \mid q\}, C = \{x \in U \mid r\}.$
(i)
\[(p \Rightarrow q \land q \Rightarrow r) \Rightarrow (p \Rightarrow r)\]
(ii)
\[p \Rightarrow q \equiv q \Rightarrow \sim p\]
4. For each of the following laws of propositional logic find a corresponding law of set theory
   (i) $p \lor \neg p \equiv \text{true}$
   (ii) $\neg(p \land q) \equiv \neg p \lor \neg q$

   Solution:
   Let $A = \{ x \in U \mid p \}$, $B = \{ x \in U \mid q \}$.
   (i) $A \cup \sim A = U$
   (ii) $\sim (A \cap B) = \sim A \cup \sim B$

5. Prove with the help of a truth table that $(p \Rightarrow q \land q \Rightarrow r) \Rightarrow (p \Rightarrow r)$ is a law of propositional logic.

   Solution:
   Let $E = (p \Rightarrow q \land q \Rightarrow r) \Rightarrow (p \Rightarrow r)$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$r$</th>
<th>$p \Rightarrow q \land q \Rightarrow r$</th>
<th>$p \Rightarrow r$</th>
<th>$E$</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>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>t</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</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>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

6. Perform the following substitutions.
   (i) $(x+x \cdot y+x \cdot y \cdot z)[z,y:=x \cdot y,x \cdot x]$  
   (ii) $(x+x \cdot y+x \cdot y \cdot z)[z:=x \cdot y][y:=x \cdot x]$  

   Solution:
   (i) $(x+x \cdot x \cdot x+x \cdot x \cdot x \cdot y)$
   (ii) $(x+x \cdot x \cdot x+x \cdot x \cdot x \cdot x \cdot x \cdot x \cdot x)$
7. Prove that the formula \((p \land (p \Rightarrow q)) \Rightarrow q\) is a theorem of propositional calculus.

Solution:
\[
(p \land (p \Rightarrow q)) \Rightarrow q
= A12

\neg(p \land (\neg p \lor q)) \lor q
= De Morgan

\neg p \lor \neg (\neg p \lor q) \lor q
= A6

\neg p \lor q \lor (\neg p \lor q)
= A10

true

8. Draw a circuit diagram corresponding to the following boolean expression \((a \lor b) \land \neg c\). Use only n-and gates.

Solution:
\[(a \lor b) \land \neg c\]

= Double Negation

\neg \neg ((a \lor b) \land \neg c)

= De Morgan

\neg \neg (\neg (a \land \neg b) \land \neg c)

9. Implement function F described below using just one return statement.

```c
bool F(int x)
{
    if(x%400==0)
        return true;
    else if(x%100==0)
        return false;
    else if(x%4==0)
        return true;
    else
        return false;
}
```

```c
bool F(int x)
{
    return ???;
}
```

**Solution:**
```c
bool F(int x)
{
    return x%400==0 || (x%4==0 && x%100!=0);
}
```

10. Prove the following formulas. Indicate which quantification rules you use.

   (i) \((\forall i \leq i \leq m-1:(\forall j \leq j \leq n-1: x_{ij} = i + j)) = \( (\forall j \leq j \leq n-1:(\forall i \leq i \leq m-1: x_{ij} = i + j)) \)

   Solution:
   (i) Apply Nesting (ii)

   (ii) \((i \leq i \leq n-1 \land 2 \land i: x_{i} + i) \cdot (i \leq i \leq n-1 \land \neg (2 \land i: x_{i} + i)) = \( (i \leq i \leq n-1: x_{i} + i) \)

   Solution:
   (ii) Apply Range Split (ii)
11. Prove that formula \( (\exists x \mid R \land P) \equiv (\exists x \mid R \land P : true) \) is a theorem of predicate calculus.
Solution:
\[
(\exists x \mid R \land P) \\
= \langle Thm5 \rangle \\
(\exists x \mid R \land P) \\
= \langle q \equiv q \land true \rangle \\
(\exists x \mid R \land P \land true) \\
= \langle Thm5 \rangle \\
(\exists x \mid R \land P : true)
\]

12. An integer type array \( x_{ij}, 0 \leq i \leq m-1, 0 \leq j \leq n-1 \) is given.
   (i) Write a quantification expression, which takes care of the following operations. Add the elements of each column with an even index. Multiply the elements of each column with an odd index. Add all the numbers you have obtained.
   (ii) Write a C++ function, which performs operations specified in (i).
Solution:
\[
(+j \mid 0 \leq j \leq n-1 \land 2 \mid j : (+i \mid 0 \leq i \leq m-1 : x_{ij})) \\
+(+j \mid 0 \leq j \leq n-1 \land -(2 \mid j : (\star i \mid 0 \leq i \leq m-1 : x_{ij}))
\]

```cpp
int F(int** x, int n, int m)
{
    int s, p, u, v, i, j;
    u = 0;
    v = 0;
    for (j = 0; j < n; j = j + 2)
    {
        s = 0;
        for (i = 0; i < m; i++)
            s = s + x[i][j];
        u = u + s;
    }
}
```
for(j=1;j<n;j=j+2)
{
    p=1;
    for(i=0;i<m;i++)
        p=p*x[i][j];
    v=v+p;
}
return u+v;

13. Prove by induction that
\[(+i|0 \leq i \leq n-1; 3^i) = \frac{3^n - 1}{2}\].

Solution:
Let \( P(n) = \{ (+i|0 \leq i \leq n-1; 3^i) = \frac{3^n - 1}{2} \} \)

\( P(1): (+i|0 \leq i \leq 1-1; 3^i) = 3^0 = 1 \)
\[ \frac{3^1 - 1}{2} = \frac{3 - 1}{2} = 1 \]

\( P(n) \Rightarrow P(n+1): (+i|0 \leq i \leq n; 3^i) \)
\[ = \text{<Range Split>} \]
\[ (+i|0 \leq i \leq n-1; 3^i) + 3^n \]
\[ = \text{<P(n)>} \]
\[ \frac{3^n - 1}{2} + 3^n \]
\[ = \text{<Arithmetic>} \]
\[ \frac{3^{n+1} - 1}{2} \]

14. Prove by induction that function \( F \) defined recursively as
\[ F(n) = \begin{cases} 
1 & \text{for } n=1 \\
2F(n-1) + 1 & \text{for } n > 1 
\end{cases} \]
satisfies
\[ F(n) = 2^n - 1. \]
Solution:
Let \( P(n) = \{ F(n) = 2^n - 1 \} \)

\[ P(1): \quad F(1) = 1 \]
\[ 2^1 - 1 = 1 \]

\[ P(n) \implies P(n+1): \quad F(n+1) \]
\[ = \text{Definition of } F \]
\[ 2F(n) + 1 \]
\[ = \text{Definition of } P \]
\[ 2(2^n - 1) + 1 \]
\[ = \text{Arithmetic} \]
\[ 2^{n+1} - 1 \]

15. We define the depth of a binary tree as the largest number of edges on the way from the root to one of the leaves. Prove by induction, that the following recursive function computes the depth of a given tree.

```c
int depth(TNode* T_ptr)
{
    if(T_ptr!=NULL)
    {
        int max;
        int left;
        int right;
        left=depth(T_ptr->left_link);
        right=depth(T_ptr->right_link);
        if(left>right)
            max=left;
        else
            max=right;
        return max+1;
    }
    else
    {
        return -1;
    }
}
```
Solution:

Proof by strong induction:

- **Recursive function** depth computes the depth of any binary tree consisting of *k* nodes.

**P(1):**

left = -1, right = -1, max = -1 + 1 = 0

The function returns 0, which is the depth of a tree consisting of a single node.

\((\forall k \mid 1 \leq k \leq n : P(k)) \Rightarrow P(n+1):\)

Let us take any binary tree consisting of *n* + 1 nodes.

The left sub-tree of the root consists of at most *n* nodes and the right sub-tree of the root consists of at most *n* nodes. In the first call we process the root. By inductive hypothesis we have

left = the depth of the left sub-tree,
right = the depth of the right sub-tree.

Therefore max + 1 is the depth of the tree we are considering.

16. Prove correctness of the following loop

\{Q: 0 \leq n\}

\(i, p := 0, 1;\)

\{P: 0 \leq i \leq n \land p = (\bullet k \mid 0 \leq k < i: k + 1)\}

do \ i \neq n \rightarrow i, p := i + 1, p \bullet (i + 1)\ od

\{R: p = (\bullet k \mid 0 \leq k < n: k + 1)\}

\Q\ the\ precondition
\P\ the\ loop\ invariant
\R\ the\ post-condition

Solution:

To prove correctness of the loop do \(B \rightarrow S\) od we need to show:

(a) \(P\) is true before execution of the loop.

(b) \(P\) is a loop invariant: \(\{P \land B\} S \{P\} .\)
(c) Execution of the loop terminates.
(d) $R$ holds upon termination: $P \land \neg B \Rightarrow R.$

(a)
\[ P[i, p := 0, 1] \]
\[ \langle \text{Substitution} \rangle \]
\[ 0 \leq 0 \leq n \land 1 = (\star k | 0 \leq k < 0 : k + 1) \]
\[ \langle \text{EmptyRangeRule} \rangle \]
\[ true \]

(b)
\[ P[i, p := i + 1, p \cdot (i + 1)] \]
\[ \langle \text{Substitution} \rangle \]
\[ 0 \leq i + 1 \leq n \land p \cdot (i + 1) = (\star k | 0 \leq k < i + 1 : k + 1) \]
\[ \langle 0 \leq i \leq n \land p = (\star k | 0 \leq k < i : k + 1) \land i \neq n \rangle \]
\[ true \]

(c)
After $n$ steps $i = n$ and the loop terminates.

(d)
\[ (0 \leq i \leq n \land p = (\star k | 0 \leq k < i : k + 1) \land i = n) \Rightarrow \]
\[ p = (\star k | 0 \leq k < n : k + 1) \]