The submission of your assignment must be done via the assignment boxes in the student foyer. Failure to submit by the due date will result in late penalties of ten percent per weekday. Other arrangements that are required because of truly exceptional circumstances need to be negotiated before your deadline.

Your submission must be well-presented on clean A4 paper with a fully completed standard cover page, including your tutor’s name and your tutorial group. Failure to follow this instruction will result in a penalty. The COMP2600 Assignments page has a link to an appropriate cover page.

1 Structural Induction [8 marks]

Exercise 1.1

\[
\text{countTrue } [] = 0 \quad \text{-- (C1)} \\
\text{countTrue } (x:xs) \\
\quad | (x == \text{True}) = 1 + \text{countTrue } xs \quad \text{-- (C2)} \\
\quad | \text{otherwise} = \text{countTrue } xs \quad \text{-- (C3)} \\
\]

\[
\text{easyCountTrue } xs = \text{length } (\text{filter } (== \text{True}) xs) \quad \text{-- (EC1)} \\
\]

\[
\text{length } [] = 0 \quad \text{-- (L1)} \\
\text{length } (x:xs) = 1 + \text{length } xs \quad \text{-- (L2)} \\
\]

\[
\text{filter } p [] = [] \quad \text{-- (F1)} \\
\text{filter } p (x : xs) \\
\quad | p x = x : \text{filter } p xs \quad \text{-- (F2)} \\
\quad | \text{otherwise} = \text{filter } p xs \quad \text{-- (F3)} \\
\]

Prove by structural induction the following property about the functions \text{easyCountTrue} and \text{countTrue}:

\[
\text{easyCountTrue } xs = \text{countTrue } xs
\]

State clearly what property \( P \) is being proved by induction, including any quantifiers needed in the statement of \( P \) and in the inductive hypothesis.
Exercise 1.2

\begin{align*}
\text{count } \text{Nul} &= 0 \quad \text{(C1)} \\
\text{count } (\text{Node } a \ t1 \ t2) &= 1 + \text{count } t1 + \text{count } t2 \quad \text{(C2)} \\
\text{counta } t &= \text{countb } t \ 0 \quad \text{(CA)} \\
\text{countb } \text{Nul} \ \text{acc} &= \text{acc} \quad \text{(CB1)} \\
\text{countb } (\text{Node } a \ t1 \ t2) \ \text{acc} &= \text{countb } t1 (1 + (\text{countb } t2 \ \text{acc})) \quad \text{(CB2)}
\end{align*}

Here is the usual Haskell definition of a binary tree:

\begin{verbatim}
data Tree a = Nul | Node a (Tree a) (Tree a)
\end{verbatim}

Prove by structural induction the following property about the functions count and counta:

\begin{align*}
\text{count } t &= \text{counta } t
\end{align*}

State clearly what property \( P \) is being proved by induction, including any quantifiers needed in the statement of \( P \) and in the inductive hypothesis.

2 FOL Specification [4 marks]

Exercise 2.1

Use the predicates \( S(x) - x \) is succesful, \( F(x) - x \) fails, \( C - \) the transaction is committed, to translate the following sentences into first-order logic:

i. The transaction is committed unless some operation fails

ii. The transaction is committed if every operation succeeds

Exercise 2.2

Use the predicates \( S(x) - x \) is succesful, \( F(x) - x \) fails, \( C(x) - x \) is committed, \( O(x, y) - y \) belongs to \( x \), to translate the following sentences into first-order logic:

i. Any transaction is committed unless some operation belonging to it fails

ii. Any transaction is committed if all operations belonging to it succeed

Compare the statements given here to those in 2.1. That is, compare (2.1.i) to (2.2.i) and (2.1.ii) to (2.2.ii). Which version is more likely to be useful when describing a realistic system, and why?

3 Truth Tables [1 mark]

Exercise 3.1

Use truth-tables to prove or disprove whether the proposition \(((a \rightarrow b) \land (b \rightarrow c)) \rightarrow (a \rightarrow c)\) is a tautology.
4 Natural Deduction [7 marks]

Exercise 4.1
Prove the following derived rule using natural deduction:

\[ \neg (\neg p \lor q) \]

\[ \frac{\quad p}{\quad} \]

You may only use the rules given in the Appendix. Do not use algebraic laws, or any of the derived rules obtained in lectures.

Exercise 4.2
Prove the following derived rule using natural deduction:

\[ \exists x. (P(x) \lor \neg Q(x)) \]

\[ \frac{\forall y. Q(y)}{\exists x. P(x)} \]

In addition to the rules given in the Appendix, you may use the following rule:

\[ p \land \neg p \]

\[ q \]

(contradiction)

Appendix 1 — Truth Table Values

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p \lor q</th>
<th>p \land q</th>
<th>p \rightarrow q</th>
<th>\neg p</th>
<th>p \leftrightarrow q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</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>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Appendix 2 — Natural Deduction Rules

\[(\land I) \quad \frac{p \quad q}{p \land q} \quad (\land E) \quad \frac{p \land q}{p} \quad \frac{p \land q}{q}\]

\[(\lor I) \quad \frac{p}{p \lor q} \quad \frac{p}{q \lor p} \quad (\lor E) \quad \frac{p \lor q \quad r \quad r}{r}\]

\[(\to I) \quad \frac{q}{p \to q} \quad (\to E) \quad \frac{p \quad p \to q}{q}\]

\[(\neg I) \quad \frac{q \land \neg q}{\neg p} \quad (\neg E) \quad \frac{q \land \neg q}{p}\]

\[(\forall I) \quad \frac{P(a) \quad (a \text{ arbitrary})}{\forall x. P(x)}\]

\[(\forall E) \quad \frac{\forall x. P(x)}{P(a)}\]

\[(\exists I) \quad \frac{P(a)}{\exists x. P(x)}\]

\[(\exists E) \quad \frac{\exists x. P(x) \quad q \quad (a \text{ arbitrary})}{q \quad (a \text{ is not free in } q)}\]