CS 511 Formal Methods, Fall 2024	Instructor: Assaf Kfoury

Homework Assignment 01

September 12, 2024 — Lucas Miguel Tassis

Exercise 1

Exercise 1.2.1 – (h): $p \vdash (p \to q) \to q$

1.	p	premise
2.	$p \to q$	assumption
3.	q	$\to$e 1, 2
4.	$(p \to q) \to q$	$\to$i 2–3

Exercise 1.2.1 – (i): $(p \to r) \wedge (q \to r) \vdash p \wedge q \to r$

1.	$(p \to r) \wedge (q \to r)$	premise
2.	$p \to r$	$\wedge e_1$ 1
3.	$p \wedge q$	assumption
4.	p	$\wedge e_1$ 3
5.	r	$\to$e 2, 4
6.	$(p \wedge q) \to r$	$\to$i 3–5

Exercise 1.2.1 – (j): $q \to r \vdash (p \to q) \to (p \to r)$

1.	$q \to r$	premise
2.	$p \to q$	assumption
3.	p	assumption
4.	q	$\to$e 2, 3
5.	r	$\to$e 1, 4
6.	$p \to r$	$\to$i 3–5
7.	$(p \to q) \to (p \to r)$	$\to$i 2–6

Exercise 2

Exercise 1.4.2 – (g): $((p \to q) \to p) \to p$

p	q	$p \to q$	$(p \to q) \to p$	$((p \to q) \to p) \to p$
T	**T**	**T**	**T**	**T**
T	**F**	**F**	**T**	**T**
F	**T**	**T**	**F**	**T**
F	**F**	**T**	**F**	**T**

Exercise 1.4.2 – (h): $((p \vee q) \to r) \to ((p \to r) \vee (q \to r))$

p	q	r	$p \vee q$	$p \to r$	$q \to r$	$(p \vee q) \to r$	$(p \to r) \vee (q \to r)$	$((p \vee q) \to r) \to ((p \to r) \vee (q \to r))$
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	**T**	**F**	**T**	**F**	**F**	**F**	**F**	**T**
T	**F**	**F**	**T**	**F**	**T**	**F**	**T**	**T**
T	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
F	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
F	**F**	**T**	**F**	**T**	**T**	**T**	**T**	**T**
F	**T**	**F**	**T**	**T**	**F**	**F**	**T**	**T**
F	**F**	**F**	**F**	**T**	**T**	**T**	**T**	**T**

Exercise 1.4.2 – (i): $(p \to q) \to (\neg p \to \neg q)$

p	q	$\neg p$	$\neg q$	$p \to q$	$\neg p \to \neg q$	$(p \to q) \to (\neg p \to \neg q)$
T	**T**	**F**	**F**	**T**	**T**	**T**
T	**F**	**F**	**T**	**F**	**T**	**T**
F	**T**	**T**	**F**	**T**	**F**	**F**
F	**F**	**T**	**T**	**T**	**T**	**T**

Problem 1

Exercise 1.5.3 – (b)

Let us suppose that C contains neither $\neg$ nor $\bot$. Now let us consider a formula ϕ that contains the connectives $\{\wedge, \vee, \to\}$. If every atom of ϕ is **T**, then ϕ will be **T**. We can prove this by using the truth tables of each connective in the set $\{\wedge, \vee, \to\}$ – in every single one, if all the atoms are **T**, then the formula will be **T**. The problem is that, considering only $\{\wedge, \vee, \to\}$ as the set of connectives, we cannot express the formula $\neg\phi$. Thus, C must contain $\neg$ or $\bot$.

Exercise 1.5.3 – (c)

My answer was based on this discussion.

The set $\{\leftrightarrow, \neg\}$ is not an adequate set of connectives for propositional logic. This is because we cannot create $\{\wedge, \vee, \to\}$ using only $\{\leftrightarrow, \neg\}$. We can prove this by showing that any formula using only $\{\leftrightarrow, \neg\}$ will always have an even number of **T** and **F** in the truth table, and, therefore, we cannot create connectives such as $\{\wedge, \vee, \to\}$ (that have an odd number of **T** and **F** values). If we look at the truth table of the connectives $\{\leftrightarrow, \neg\}$, we can see that we have always an even number of **T** and **F**:

ϕ	ψ	$\neg\phi$	$\neg\psi$	$\phi \leftrightarrow \neg\psi$	$\neg\phi \leftrightarrow \psi$	$\neg\phi \leftrightarrow \neg\psi$	$\phi \leftrightarrow \psi$
T	**T**	**F**	**F**	**F**	**F**	**T**	**T**
T	**F**	**F**	**T**	**T**	**T**	**F**	**F**
F	**T**	**T**	**F**	**T**	**T**	**F**	**F**
F	**F**	**T**	**T**	**F**	**F**	**T**	**T**

Thus, if ϕ and ψ are formulas only using $\{\leftrightarrow, \neg\}$, they still will always have a even number of **T** and **F**. Therefore, the set $\{\leftrightarrow, \neg\}$ is not an adequate set of connectives for propositional logic.

Exercise 3

Link to code on GitHub. The solution was:

```
-- Exercise 3:  MoP Example 1.3.4
example {w : ℚ} (h1 : 3 * w + 1 = 4) : w = 1 := by
  calc
    w = ((3 * w + 1) / 3) - (1 / 3) := by ring
    _ = (4 / 3) - 1 / 3 := by rw [h1]
    _ = 1 := by ring
```

Exercise 4

Link to code on GitHub. The solution was:

```
-- Exercise 4:  MoP Example 1.3.9
example {a b : ℚ} (h1 : a - 3 = 2 * b) : a ^ 2 - a + 3 = 4 * b ^ 2 + 10 * b + 9 := by
  calc
    a ^ 2 - a + 3 = ((a - 3) ^ 2 + 6 * a - 9) - a + 3 := by ring
    _ = (a - 3) ^ 2 + 5 * a - 6 := by ring
    _ = (a - 3) ^ 2 + 5 * ((a - 3) + 3) - 6 := by ring
    _ = (a - 3) ^ 2 + 5 * (a - 3) + 9 := by ring
    _ = (2 * b) ^ 2 + 5 * (2 * b) + 9 := by rw [h1]
    _ = 4 * b ^ 2 + 10 * b + 9 := by ring
```

Problem 2

Solving by hand:

$$
\begin{aligned}
a &= (a + 2b + 3c) - 2b - 3c \\
&= 7 - 2b - 3c \\
&= 7 - (b + 2c) - b - c \\
&= 7 - 3 - b - 1 \\
&= -b - 2c + 2c + 3 \\
&= -(b + 2c) + 2c + 3 \\
&= -3 + 2 \cdot 1 + 3 \\
&= 2
\end{aligned}
$$

Link to code on GitHub. The Lean solution was:

```
-- Problem 2:  MoP Exercise 1.3.11
example {a b c : ℝ} (h1 : a + 2 * b + 3 * c = 7) (h2 : b + 2 * c = 3)
    (h3 : c = 1) : a = 2 := by
  calc
    a = (a + 2 * b + 3 * c) - 2 * b - 3 * c := by ring
    _ = 7 - 2 * b - 3 * c := by rw [h1]
    _ = 7 - (b + 2 * c) - b - c := by ring
    _ = 7 - 3 - b - 1 := by rw [h2, h3]
    _ = - b - 2 * c + 2 * c + 3 := by ring
    _ = - (b + 2 * c) + 2 * c + 3 := by ring
```

```
_ = - 3 + 2 * 1 + 3 := by rw [h2, h3]
_ = 2 := by ring
```