COMP2711H Tutorial 2

Yuchen Mao

Department of Computer Science and Engineering,
Hong Kong University of Science and Technology

1 Proposition

At the end of the last class, we claim that $(p_1 \vee \ldots \vee p_n) \to q \equiv (p_1 \to q) \wedge \ldots \wedge (p_n \to q)$. Here I would like to offer a proof of this equivalence.

$$\begin{aligned}(p_1 \vee \ldots \vee p_n) \to q &\equiv \neg(p_1 \vee \ldots \vee p_n) \vee q\\ &\equiv (\neg p_1 \wedge \ldots \wedge \neg p_n) \vee q\\ &\equiv (\neg p_1 \vee q) \wedge \ldots \wedge (\neg p_n \vee q)\\ &\equiv (p_1 \to q) \wedge \ldots \wedge (p_n \to q)\end{aligned}$$

In the first line, we use the equivalence $p \to q \equiv \neg p \vee q$. In the second line, we use $\neg(p \vee q) \equiv \neg p \wedge \neg q$. In the third line, we use the distributive law. In the forth line, we $p \to q \equiv \neg p \vee q$.

2 Predicate

2.1 Domain

The domain can affect the correctness of a predicate formular.

Exercise 2.1. Consider the correctness of $\forall x, x > 0$ in the following domain:

1. $\mathbb{Z}$: the set of integers;
2. $\mathbb{N}$: the set of natural numbers;
3. $\mathbb{N}^+$: the set of positive natural numbers.

2.2 Negation

Exercise 2.2. Move the $\neg$ as inside as you can.

$$\neg\forall x \exists y \exists z, P(x, y, z)$$

The general principle is that moving a $\neg$ across a quantifier changes the kind of quantifier.

Exercise 2.3. Move the $\neg$ as inside as you can.

$$\neg\forall x \forall y((\exists z, P(x, y, z)) \wedge (\forall z, Q(x, y, z)))$$

2.3 Equivalence and Validity of Predicate Formular

Exercise 2.4. (a) Is $\forall x, P(x) \vee Q(x) \equiv (\forall x, P(x)) \vee (\forall x, Q(x))$ correct?

(b) Is $\exists x \forall y, P(x, y) \to \forall y \exists x, P(x, y)$ valid?

To prove that two quantified statements are not equivalent, it suffices to show a counterexample. For example, in exercise2.4(a), you may let $P(x)$ be the statement that x is male, let $Q(x)$ be x is female, and let the domian be all students in COMP2711H class. Then the left-hand side states that for any student in COMP2711H, it is either male or female, which is certainly true, while the right-hand side states that either all students in COMP2711H are male or all students in COMP2711H are female, which is false.

However, it can be subtle to show that two quantified statements are equivalent (or that a quantified statement is valid). Take exercise2.4(b) for example. $\exists x \forall y, P(x, y)$ states that there exists a x, namely x_0, such that for all y, $P(x_0, y)$ is true. Then for all y, we always have x_0 which make $P(x_0, y)$ true; this makes $\forall y \exists x, P(x, y)$ true. Hence the statement in exercise2.4(b) is valid. You can hardly call this a proof; what we have done is just restating the statement carefully in natural language.

If you really have no idea whether two quantified statementsare equivalent or nor, you may try to make those abstract statements concrete: specify the domain of the statements, give a specific meaning to the predicates in the statement, and see if they states the same thing under this specific setting. It may help you understand the quantified statement although mathematically it does nothing.

2.4 Change Domain in Quantified Statements

Exercise 2.5. Is the following quantified statement true or false?

$$\forall x \in \mathbb{R}, x < 8 \to x < 9$$

Exercise 2.6. When the domain is $\mathbb{Z}$, is the following quantified statement true or false?

$$\forall x, x < 10 \to (\forall y, y < x \to y < 9)$$

Theorem 2.1. Let $P(x), Q(x)$ be two predicates. Let U_1, U_2 be two domains where

$$U_2 = \{x \in U_1 : Q(x) \text{ is ture}\}$$

Then we have that

$$\forall x \in U_2, P(x) \equiv \forall x \in U_1, Q(x) \to P(x)$$

Theorem 2.2. Let $P(x), Q(x)$ be two predicates. Let U_1, U_2 be two domains where

$$U_2 = \{x \in U_1 : Q(x) \text{ is ture}\}$$

Then we have that

$$\exists x \in U_2, P(x) \equiv \exists x \in U_1, Q(x) \wedge P(x)$$

References

[1] HKUST. *COMP2711 Lecture Notes.*

[2] E. Lehman, T. Leighton, and A. R. Meyer. *Mathematics for computer science*, chapter 3. 2010.