Proposition - Supplementary

COMP9020 Tutorial

JIAPENG WANG

24T3 Week7, H18B & F15A

1 Tutorial Outline

1.1 Definition

Ask yourself these questions below to test your understanding.

- What is the definition of the *propositions*?
  - What is an *atomic proposition*?
  - How to translate natural language into propositions?
- What *operators* can be used to combine propositions?
  - Basic and Conditional Connectives
    - What is *truth table*?
    - What is the recursive definition of *wffs*?
    - What is the basic method for proving *logically equivalence*?
    - How to convert when we encounter *implications* ($\rightarrow$)?
  - Entailment
    - What is an *argument*?
    - What is the basic method for proving *validity* of an argument?

1.2 Brainstorming

The following questions are open and have no standard answers.

- Why can we prove propositions using *indirect methods*?
- What are the applications of *symbolic logic* in computer science?
- What are the limitations of *propositional logic*?
- ......

2 Another Way to Prove Validity

Please note that this section is merely an extension of symbolic logic and is not part of the course content. It is intended to deepen your understanding of mathematical problem analysis.

DO NOT use this method to prove arguments about validity in **this term**'s assignments or exams!

2.1 Valid Argument Form

We can always use **truth tables** based on their definitions when proving *logical equivalence* and *argument validity.*

Moreover, since we can prove the *five basic laws* using the definition of logical equivalence—thus showing that the structure $(\texttt{Prop}, \vee, \wedge, \neg, \bot, \top)$ a Boolean algebra—we can also rely on common laws for our proofs. But when it comes to argument validity, is there another method we can use for proof?

The answer is yes. We can also prove the following rules using *the definition of valid arguments.* (Proof omitted.)

Table 1: Valid Argument Form for Proposition

Modus Ponens	$p \to q,\ p \models q$
Modus Tollens	$p \to q,\ \neg q \models \neg p$
Generalization	$p \models p \vee q$ $q \models p \vee q$
Specialization	$p \wedge q \models p$ $p \wedge q \models q$
Conjunction	$p,\ q \models p \wedge q$
Elimination	$p \vee q,\ \neg q \models p$ $p \vee q,\ \neg p \models q$
Transitivity	$p \to q,\ q \to r \models p \to r$
Proof by Division into Cases	$p \vee q,\ p \to r,\ q \to r \models r$
Contradiction Rule	$\neg p \to \bot \models p$
Conditional Proof Rule	$(p \models q) \models p \to q$

Using these rules, we can more quickly prove the argument's validity. For example, consider this problem:

Exercise: (Assume all variables are statement variables.)

a. $p \to q$
b. $r \lor s$
c. $\neg s \to \neg t$
d. $\neg q \lor s$
e. $\neg s$
f. $\neg p \land r \to u$
g. $w \lor t$
$\therefore$ $u \land w$

Proof:

(1)	$\neg s \to \neg t$	by premise (c)
	$\neg s$	by premise (e)
	$\therefore \neg t$	by modus ponens
(2)	$w \lor t$	by premise (g)
	$\neg t$	by (1)
	$\therefore w$	by elimination
(3)	$\neg q \lor s$	by premise (d)
	$\neg s$	by premise (e)
	$\therefore \neg q$	by elimination
(4)	$p \to q$	by premise (a)
	$\neg q$	by (3)
	$\therefore \neg p$	by modus tollens
(5)	$r \lor s$	by premise (b)
	$\neg s$	by premise (e)
	$\therefore r$	by elimination
(6)	$\neg p$	by (4)
	r	by (5)
	$\therefore \neg p \land r$	by conjunction
(7)	$\neg p \land r \to u$	by premise (f)
	$\neg p \land r$	by (6)
	$\therefore u$	by modus ponens
(8)	u	by (7)
	w	by (2)
	$\therefore u \land w$	by conjunction

If we draw its *truth table* and examine the *critical row*, we can also see that the argument is valid; however, this method is far less concise.

This reasoning approach may seem familiar in this proof—does it remind you of something? Indeed, **ALL** the analyses we've done in lectures, tutorials, and quizzes follow this format, except in those examples, $p, q, r, s, \ldots$ are no longer mere propositional variables but actual propositions.

For example, suppose we want to prove that

`Prop1.` If n is odd, then $n - 1$ is even.

We begin by assigning propositions as follows:

- p: n is an odd number.
- q: $n - 1$ is an even number.

Thus, the proof is equivalent to showing $p \models q$. However, we cannot directly reach this conclusion based on these two conditions alone. Therefore, we need to identify some universally accepted facts, such as the **definitions** of odd and even numbers:

- n is odd if and only if there exists an integer k such that $n = 2k + 1$;
- n is even if and only if there exists an integer k such that $n = 2k$.

Based on this, we introduce additional propositions:

- r: n can be expressed as $2k + 1$ for some integer k.
- s: $n - 1$ can be expressed as $2k$ for some integer k.

According to these definitions, we have $p \to r$ and $s \to q$. [1]

Furthermore, through *equation manipulation*, we find that $r \to s$ also holds. Now, this argument can be stated as:

a. p

b. $p \to r$

c. $s \to q$

d. $r \to s$

$\therefore q$

Through analysis, we conclude that these premises are sufficiently comprehensive, allowing us to derive the conclusion using **Modus Ponens**. Hence, we complete the analysis and proof of this problem.

Without even realizing it, we've already been following these rules to analyze and solve problems, haven't we? By translating these principles into symbols, we can more intuitively grasp the underlying logic and explore more abstract and universal problems. This is precisely the advantage of *symbolic logic*.

[1]In fact, they should all be $\leftrightarrow$, but in the proof for this problem, we only need a single direction.

2.2 Inference Rules for Predicate

However, due to the inherent limitations of propositional logic, we cannot use those forms to check the validity of all arguments; hence, we introduce **predicate logic**. This leads to the development of additional rules (Quantifier Elimination and Introduction), allowing us to better handle universal and existential statements[2]:

Table 2: Inference Rules for Predicate

UQ Elim	$\forall x P(x) \models P(c)$ for an arbitrary c
EQ Elim	$\exists x P(x) \models P(c)$ for some element c
UQ Intro	$P(c)$ for an arbitrary $c \models \forall x P(x)$
EQ Intro	$P(c)$ for some element $c \models \exists x P(x)$

Now, if we consider the proposition (universal statement):

`Prop2.` For all odd numbers n, $n - 1$ is even.

Let predicate $P(n), Q(n)$ be:
- $P(x)$: x are odd numbers.
- $Q(x)$: x are even numbers.

We can first apply UQElim to transform $\forall n\ P(n)$ into $P(c)$ (for an arbitrary c). This brings us back to `Prop1` demonstrated above. After proving `Prop1`, we get the $Q(c-1)$ (for an arbitrary c). Then use UQIntro to revert to the conclusion $\forall n\ Q(n-1)$, thus completing the proof.

2.3 Summary

If you look back at all the analysis I've done in the tutorials, you'll notice they all follow this approach essentially. The purpose of this course is

- To increase your level of mathematical maturity to help with the fundamental task of **discovering**, **formulating**, and **proving** properties of programs.[3]

Therefore, I believe it is necessary to introduce you to these analysis principles. Think back to what I mentioned in our first tutorial that

- The parts of *writing proofs* are better discussed after learning propositional logic.

Do you see what I meant now? `;)`

[2]Note that all propositional logic p should now be expressed as predicate logic $P(x)$.
[3]Slides of Lecture1, P6