1.6 Rules of Inference

**Valid Arguments in Propositional Logic**

**Definition 1.6.1.** An *argument* in propositional logic is a sequence of propositions. All but the final proposition in the argument are called *premises* and the final proposition is called the *conclusion*. An argument is *valid* if the truth of all its premises implies that the conclusion is true. An *argument form* in propositional logic is a sequence of compound propositions involving propositional variables. An argument form is *valid* no matter which particular propositions are substituted for the propositional variables in its premises, the conclusion is true if the premises are all true.

**Remark 1.6.2.** From the definition of a valid argument form we see that the argument form with premises $p_1, p_2, \cdots, p_n$ and conclusion $q$ is valid, when $(p_1 \land p_2 \land \cdots \land p_n) \rightarrow q$ is a tautology. The key to showing that an argument in propositional logic is valid is to show that its argument form is valid.

**Example 1.6.3.** 1. Consider the following argument involving propositions:

"If you have a current password, then you can log onto the network."

"You have a current password."

Therefore, "You can log onto the network."

We would like to determine whether this is a valid argument. That is, we would like to determine whether the conclusion "You can log onto the network" must be true when the premises "If you have a current password, then you can log onto the network" and "You have a current password" are both true.

Use $p$ to represent "You have a current password" and $q$ to represent "You can log onto the network." Then, the argument has the form:
2. Consider the following argument involving propositions:

"If you have access to the network, then you can change your grade."
"You have access to the network."
Therefore "You can change your grade."

Remark 1.6.4. In our discussion, to analyze an argument, we replaced propositions by propositional variables. This changed an argument to an argument form. We saw that the validity of an argument follows from the validity of the form of the argument.

**Rules of Inference for Propositional Logic**

- We can always use a truth table to show that an argument form is valid. We do this by showing that whenever the premises are true, the conclusion must also be true.

- The truth table can be a tedious approach. For example, when an argument form involves 10 different propositional variables, to use a truth table to show this argument form is valid requires $2^{10} = 1024$ different rows.

- We do not have to resort to truth tables. Instead, we can first establish the validity of some relatively simple argument forms, called *rules of inference*.

- These rules of inference can be used as building blocks to construct more complicated valid argument forms.

We will now introduce the most important rules of inference in propositional logic. The tautology $(p \land (p \rightarrow q)) \rightarrow q$ is the basis of the rule of inference called *modus ponens*, or the *law of detachment*. This tautology leads to the following valid argument form, which we have already seen in our initial discussion about arguments (where, as before, the symbol $\therefore$ denotes
therefore”):

\[ P1: \quad p \]

\[ P2: \quad p \rightarrow q \]

\[ C: \quad \therefore q \]

Using this notation, the hypotheses are written in a column, followed by a horizontal bar, followed by a line that begins with the therefore symbol and ends with the conclusion.

**Example 1.6.5.** Suppose that the conditional statement "If it snows today, then we will go skiing" and its hypothesis, "It is snowing today," are true. Then, by modus ponens, it follows that the conclusion of the conditional statement, "We will go skiing," is true.

As we mentioned earlier, a valid argument can lead to an incorrect conclusion if one or more of its premises is false. We illustrate this again in the following example:

**Example 1.6.6.** Determine whether the argument given here is valid and determine whether its conclusion must be true because of the validity of the argument. "If \( \sqrt{2} > 2 \), then \( (\sqrt{2})^2 > 2^2 \)." We know that \( \sqrt{2} > 2 \). Consequently, \( (\sqrt{2})^2 > 2^2 \).

There are many useful rules of inference for propositional logic. Perhaps the most widely used of these are listed in the following table:


<table>
<thead>
<tr>
<th>Example 1.6.7. State which rule of inference is the basis of the following arguments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. &quot;It is below freezing now. Therefore, it is either below freezing or raining now.&quot;</td>
</tr>
<tr>
<td>2. &quot;If you study computer science, then you get a job. If you get a job, then you are suc-</td>
</tr>
</tbody>
</table>
Using Rules of Inference to Build Arguments

- When there are many premises, several rules of inference are often needed to show that an argument is valid. This is illustrated by the following examples where the steps of arguments are displayed on separate lines, with the reason for each step explicitly stated.

- These examples also show how arguments in English can be analyzed using rules of inference.

Example 1.6.8. 1. Determine whether the following argument is a valid argument:

\[ W \lor (H \land L) \]

\[(W \lor H) \rightarrow D/ :. W \lor D.\]
2. Use rules of inference to show that the hypotheses imply the conclusion:

"Ahmed works hard."

"If Ahmed works hard, then he is a good boy."

"If Ahmed is a good boy, then he will get the job."

"Therefore Ahmed will get the job."

3. Show that the premises "It is not sunny this afternoon and it is colder than yesterday,

"We will go swimming only if it is sunny,"

"If we do not go swimming, then we will take a canoe trip,

"If we take a canoe trip, then we will be home by sunset"

lead to the conclusion "We will be home by sunset."

4. Show that the premises "If you send me an e-mail message, then I will finish writing the program," "If you do not send me an e-mail message, then I will go to sleep early," and "If I go to sleep early, then I will wake up feeling refreshed" lead to the conclusion "If I do not finish writing the program, then I will wake up feeling refreshed."
Remark 1.6.9. For the third example, note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, \( p, q, r, s, \) and \( t \), such a truth table would have 32 rows.

### Resolution

- Computer programs have been developed to automate the task of reasoning and proving theorems. Many of these programs make use of a rule of inference known as resolution. This rule of inference is based on the tautology \( ((p \lor q) \land (\neg p \lor r)) \rightarrow (q \lor r) \). The final disjunction in the resolution rule, \( q \lor r \), is called the resolvent.

- When we let \( q = r \) in this tautology, we obtain \( (p \lor q) \land (\neg p \lor q) \rightarrow q \).

- When we let \( r \) to be a contradiction \( r = c \), we obtain \( (p \lor q) \land (\neg p) \rightarrow q \) (because \( q \lor c \equiv q \)), which is the tautology on which the rule of disjunctive syllogism is based.

**Example 1.6.10.** Use resolution to show that the hypotheses "Ali is skiing or it is not snowing" and "It is snowing or Ahmed is playing football" imply that "Ali is skiing or Ahmed is playing football."

**Definition 1.6.11.** A clause is a disjunction of variables or negations of these variables.

### Resolution and programming languages

- Resolution plays an important role in programming languages based on the rules of logic. It can be used to build automatic theorem proving systems. To construct proofs in propositional logic using resolution as the only rule of inference, the hypotheses and the conclusion must be expressed as clauses.

- We can replace a statement in propositional logic that is not a clause by one or more equivalent statements that are clauses. For example:
1. The form $p \lor (q \land r)$. Since $p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$, we can replace the single statement $p \lor (q \land r)$ by two statements $p \lor q$ and $pr$, each of which is a clause.

2. By De Morgan's law, $\neg(p \lor q) \equiv \neg p \land \neg q$. So the form $\neg(p \lor q)$ can be replaced by the two statements $\neg p$ and $\neg q$.

3. We can also replace a conditional statement $p \rightarrow q$ with the equivalent disjunction $\neg p \lor q$.

Example 1.6.12. Show that the premises $(p \land q) \lor r$ and $r \rightarrow s$ imply the conclusion $p \lor s$.

Fallacies

Several common fallacies arise in incorrect arguments. These fallacies resemble rules of inference, but are based on contingencies rather than tautologies. These are discussed here to show the distinction between correct and incorrect reasoning. The proposition $((p \rightarrow q) \land q) \rightarrow p$ is not a tautology, because it is false when $p$ is false and $q$ is true. However, there are many incorrect arguments that treat this as a tautology. Also the proposition $((p \rightarrow q) \land \neg p) \rightarrow \neg q$ is not a tautology.

Example 1.6.13. Is the following arguments valid?

1. "If you do every problem in this book, then you will learn discrete mathematics.” "You learned discrete mathematics.” "Therefore, you did every problem in this book.”

2. "If you do every problem in this book, then you will learn discrete mathematics.” "You did not do every problem in the book.” "Therefore you did not learn discrete mathematics”
Rules of Inference for Quantified Statements

We will now describe some important rules of inference for statements involving quantifiers. These rules of inference are used extensively in mathematical arguments, often without being explicitly mentioned.

The following table gives rules of inference for quantified statements.

<table>
<thead>
<tr>
<th>Rule of Inference</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \forall x P(x) ) [ \vdash ] P(c)</td>
<td>Universal instantiation</td>
</tr>
<tr>
<td>( P(c) ) for an arbitrary c [ \vdash ] ( \forall x P(x) )</td>
<td>Universal generalization</td>
</tr>
<tr>
<td>( \exists x P(x) ) [ \vdash ] P(c) for some element c</td>
<td>Existential instantiation</td>
</tr>
<tr>
<td>( P(c) ) for some element c [ \vdash ] ( \exists x P(x) )</td>
<td>Existential generalization</td>
</tr>
</tbody>
</table>

**Universal instantiation** is the rule of inference used to conclude that \( P(c) \) is true, where \( c \) is a particular member of the domain, given the premise \( (\forall x)P(x) \). Universal instantiation is used when we conclude from the statement "All students are tall" that "Mohammed is tall," where Mohammed is a member of the domain of all students.

**Existential generalization** Universal generalization is the rule of inference that states that \( (\forall x)P(x) \) is true, given the premise that \( P(c) \) is true for all elements \( c \) in the domain. Universal generalization is used when we show that \( (\forall x)P(x) \) is true by taking an arbitrary element \( c \) from the domain and showing that \( P(c) \) is true. The element \( c \) that we select must be an arbitrary, and not a specific, element of the domain.

**Existential instantiation** is the rule that allows us to conclude that there is an element \( c \) in the domain for which \( P(c) \) is true if we know that \( (\exists x)P(x) \) is true. We cannot select an arbitrary value of \( c \) here, but rather it must be a \( c \) for which \( P(c) \) is true. Usually we have no knowledge of what \( c \) is, only that it exists. Because it exists, we may give it a name \( (c) \) and continue our argument.
Existential generalization is the rule of inference that is used to conclude that \((\exists x)P(x)\) is true when a particular element \(c\) with \(P(c)\) true is known. That is, if we know one element \(c\) in the domain for which \(P(c)\) is true, then we know that \((\exists x)P(x)\) is true.

Example 1.6.14. 1. Show that the premises "Everyone in this discrete mathematics class has taken a course in computer science" and "Nour is a student in this class" imply the conclusion "Nour has taken a course in computer science."

2. Show that the premises "A student in this class has not read the book," and "Everyone in this class passed the first exam" imply the conclusion "Someone who passed the first exam has not read the book."