Discrete Mathematics

Topic 01 — Logic

Lecture 04 — Rules of Inference

Dr Kieran Murphy

Department of Computing and Mathematics,
Waterford IT.
(kmurphy@wit.ie)

Autumn Semester, 2021

Outline

- Constructing arguments in propositional logic
- Normal forms

Outline

Notation

Single-line vs Double-line Arrows

For the purpose of this module the single line arrows (representing the IFTHEN and IFF connectives)

$$\rightarrow \qquad \text{and} \qquad \leftrightarrow$$

mean the same thing as the corresponding double-line arrow

$$\Rightarrow \qquad \text{and} \qquad \Leftrightarrow$$

I will use the double-lined arrows in places where I want to treat a complicated proposition as two smaller propositions. For example, I want to think of the proposition

$$(p \rightarrow q) \wedge \neg q \implies \neg p$$

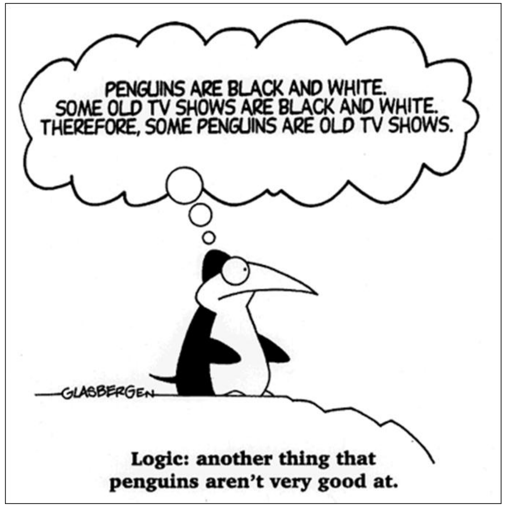
in terms of the two proposition $(p \rightarrow q) \wedge \neg q$ and $\neg p$.

Motivation

Remember the Socrates example when we started Logic.

"All men are mortal.
Socrates is a man.
Therefore, Socrates is mortal."

Compare with this argument...



Here we have two premises:

- All men are mortal
- Socrates is a man.

and the conclusion:

- Socrates is mortal.

Q: How do we get the conclusion from the premises?

A: We construct an argument, a sequence of propositions that follow from the rules of inference until we reach the conclusion.

Arguments

Definition 1 (Argument)

A argument in propositional logic is a sequence of propositions. All but the final proposition are called premises. The last statement is the conclusion. The argument is valid if the premises imply the conclusion.

- If the premises are $p_1, p_2, \ldots p_n$ and the conclusion is q then the argument is valid iff

$$(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \to q$$

  is a tautology.
- We could use truth tables to test if an argument is valid — construct the above expression, then build the truth table and check the output column.
- Alternatively, we could sequently apply inference rules to arrive at the conclusion.
- Inference rules are simple arguments that will be used to construct more complex argument forms.

Outline

Detachment (Modus Ponens)

Argument

$$\begin{array}{l} p \to q \\ p \\ \hline \therefore q \end{array}$$

Corresponding Tautology

$$(p \to q) \land p \implies q$$

Example

Let

$$p = \text{“\textit{It is snowing.}”}$$

$$q = \text{“\textit{I will study discrete maths.}”}$$

Then the argument is

“If it is snowing, then I will study discrete maths.”
“It is snowing.”

Therefore *“I will study discrete maths.”*

Indirect Reasoning (Modus Tollens)

Argument

$$\begin{array}{c} p \to q \\ \neg q \\ \hline \therefore \neg p \end{array}$$

Corresponding Tautology

$$(p \to q) \wedge \neg q \implies \neg p$$

Example

Let

$p =$ "*It is snowing.*"

$q =$ "*I will study discrete maths.*"

Then the argument is

"*If it is snowing, then I will study discrete maths.*"
"*I will not study discrete maths.*"

Therefore "*It is not snowing.*"

Chain Rule (Hypothetical Syllogism)

Argument

$$\begin{array}{l} p \to q \\ q \to r \\ \hline \therefore p \to r \end{array}$$

Corresponding Tautology

$$(p \to q) \wedge (q \to r) \implies (p \to r)$$

Example

Let

$p =$ "*It is snowing.*"

$q =$ "*I will study discrete maths.*"

$r =$ "*I will get an A.*"

Then the argument is

"*If it is snowing, then I will study discrete maths.*"

"*If I will study discrete maths, then I will get an A.*"

Therefore "*If it is snowing, then I will get an A.*"

Chain Rule (Hypothetical Syllogism)

Argument

$$\begin{array}{l} p \to q \\ q \to r \\ \hline \therefore p \to r \end{array}$$

Corresponding Tautology

$$(p \to q) \wedge (q \to r) \implies (p \to r)$$

Example

Let

$p =$ "*It is snowing.*"

$q =$ "*I will study discrete maths.*"

$r =$ "*I will get an A.*"

Then the argument is

"*If it is snowing, then I will study discrete maths.*"
"*If I will study discrete maths, then I will get an A.*"

Therefore "*If it is snowing, then I will get an A.*"

Disjunctive Simplification (Disjunctive Syllogism)

Argument

$$\begin{array}{c} p \vee q \\ \neg p \\ \hline \therefore q \end{array} \qquad \begin{array}{c} p \vee q \\ \neg q \\ \hline \therefore p \end{array}$$

Corresponding Tautology

$$(p \vee q) \wedge (\neg p) \implies q$$

$$(p \vee q) \wedge (\neg q) \implies p$$

Example

Let

$p =$ "*I will study discrete maths.*"

$q =$ "*I will study programming.*"

Then the argument is

"*I will study discrete maths or I will study programming.*"

"*I will not study discrete maths.*"

Therefore "*I will study programming.*"

Disjunctive Addition

Argument

$$\frac{p}{\therefore p \vee q}$$

Corresponding Tautology

$$p \implies (p \vee q)$$

Example

Let

$p =$ "*I will study discrete maths.*"

$q =$ "*I will get high.*"

Then the argument is

"*I will study discrete maths.*"

Therefore "*I will study discrete maths or I will get high.*"

Conjunctive Simplification

Argument

$$\frac{p \wedge q}{\therefore p} \qquad \frac{p \wedge q}{\therefore q}$$

Corresponding Tautology

$$(p \wedge q) \implies p$$
$$(p \wedge q) \implies q$$

Example

Let

$p =$ "*I will study discrete maths.*"

$q =$ "*I will get high.*"

Then the argument is

"*I will study discrete maths and I will get high.*"

Therefore "*I will study discrete maths.*"

Resolution

Argument

$$\begin{array}{l} \neg p \vee r \\ p \vee q \\ \hline \therefore q \vee r \end{array}$$

Corresponding Tautology

$$(\neg p \vee r) \wedge (p \vee q) \implies (q \vee r)$$

Example

Let

$p =$ "*I will study discrete maths.*"

$p =$ "*I will study programming.*"

$p =$ "*I will study databases.*"

Then the argument is

"*I will not study discrete maths or I will study programming.*"

"*I will study discrete maths or I will study databases.*"

Therefore "*I will study programming or I will study databases.*"

Outline

Example 2 I

A valid argument is a sequence of statements. Each statement is either a premise or follows from previous statements by rules of inference. The last statement is called conclusion.

Example 2

Assuming the following two propositions

$$p \qquad \text{and} \qquad (p \to q)$$

show that q is a conclusion.

Method 1

Construct argument using inference rules …

	Step	**Reason**
1)	$p \wedge (p \to q)$	Premise
2)	p	Conjunctive Simplification from (1)
3)	$p \to q$	Conjunctive Simplification from (1)
$\therefore$	q	Detachment (Modus Ponens) from (2) and (3)

Example 2 II

Method 2

Construct an expression of the form

$$(\text{premise } 1) \wedge (\text{premise } 2) \wedge \cdots \wedge (\text{premise } n) \implies (\text{conclusion})$$

and verify that the expression is a tautology (using a truth table).
So for this example ...

$$\underbrace{(\text{premise } 1)}_{p} \wedge \underbrace{(\text{premise } 2)}_{(p \to q)} \underbrace{\implies (\text{conclusion})}_{q}$$

inputs		individual premises		argument premise	argument conclusion	argument
p	q	p	$(p \to q)$	$p \wedge (p \to q)$	q	$p \wedge (p \to q) \Rightarrow q$
F	**F**	**F**	**T**	**F**	**F**	**T**
F	**T**	**F**	**T**	**F**	**T**	**T**
T	**F**	**T**	**F**	**F**	**F**	**T**
T	**T**	**T**	**T**	**T**	**T**	**T**

Example 3 I

Example 3

With these hypotheses:

a) *"It is not sunny this afternoon and it is colder than yesterday."*
b) *"We will go swimming only if it is sunny."*
c) *"If we do not go swimming, then we will take a canoe trip."*
d) *"If we take a canoe trip, then we will be home by sunset."*

Using the inference rules, construct a valid argument for the conclusion:

- *"We will be home by sunset."*

General procedure . . .

STEP 1 Choose propositional variables.

STEP 2 Translation into propositional logic.

STEP 3 Construct the valid argument (OR verify related tautology using truth table.)

Example 3 II

STEP 1 *Choose propositional variables.*

- s = "*It is Sunny this afternoon.*"
- c = "*It is Colder than yesterday.*"
- w = "*We will go sWimming*"
- t = "*We will take a canoe Trip.*"
- h = "*We will be Home by sunset.*"

STEP 2 *Translation into propositional logic.*

Premises ...

a) "*It is not sunny this afternoon and it is colder than yesterday.*" $\neg s \wedge c$
b) "*We will go swimming only if it is sunny.*" $w \rightarrow s$
c) "*If we do not go swimming, then we will take a canoe trip.*" $\neg w \rightarrow t$
d) "*If we take a canoe trip, then we will be home by sunset.*" $t \rightarrow h$

and conclusion

- "*We will be home by sunset.*" h

Example 3 III

STEP 3 *Construct the valid argument* *(Note the truth table here would have 32 rows)*

We have

Premises				Conclusion
(a)	(b)	(c)	(d)	
$\neg s \wedge c$	$w \to s$	$\neg w \to t$	$t \to h$	h

And our argument is ...

	Step	**Reason**
1)	$\neg s \wedge c$	Premise (a)
2)	$\neg s$	Conjunctive Simplification from (1)
3)	$w \to s$	Premise (b)
4)	$\neg w$	Indirect Reasoning (Modus Tollens) from (2) and (3)
5)	$\neg w \to t$	Premise (c)
6)	t	Detachment (Modus Ponens) from (4) and (5)
7)	$t \to h$	Premise (d)
$\therefore$	h	Detachment (Modus Ponens) from (6) and (7)

That was a bit painful . . . let Python do the work . . . I

Premises				Conclusion
(a)	(b)	(c)	(d)	
$\neg s \wedge c$	$w \rightarrow s$	$\neg w \rightarrow t$	$t \rightarrow h$	h

```python
# individual premises
p1 = "not s and c"
p2 = "not w or s"
p3 = "w or t"
p4 = "not t or h"

# construct argument premise - each premise is inside ( )
p = f"({p1}) and ({p2}) and ({p3}) and ({p4})"

# argument conclusion
c = "h"

# output argument premise and conclusion
print(f"argument premise: {p}")
print(f"argument conclusion: {c}")

# build expression for testing (is it a tautology?)
argument = f"(not ({p})) or ({c})"

# generate truth table - show premises, conclusion and argument
TruthTable([p1,p2,p3,p4, c, argument])
```

That was a bit painful . . . let Python do the work . . . II

```
argument premise: (not s and c) and (not w or s) and (w or t) and (not t or h)
argument conclusion: h
```

c	h	s	t	w	not s and c	not w or s	w or t	not t or h	h	(not ((not s and c) and (not w or s) and (w or t) and (not t or h))) or (h)
False	False	False	False	False	False	True	False	True	False	True
False	False	False	False	True	False	False	True	True	False	True
False	False	False	True	False	False	True	True	False	False	True
False	False	False	True	True	False	False	True	False	False	True
False	False	True	False	False	False	True	False	True	False	True
False	False	True	False	True	False	True	True	True	False	True
False	False	True	True	False	False	True	True	False	False	True
False	False	True	True	True	False	True	True	False	False	True
False	True	False	False	False	False	True	False	True	True	True
False	True	False	False	True	False	False	True	True	True	True
False	True	False	True	False	False	True	True	True	True	True
False	True	False	True	True	False	False	True	True	True	True
False	True	True	False	False	False	True	False	True	True	True
False	True	True	False	True	False	True	True	True	True	True
False	True	True	True	False	False	True	True	True	True	True
False	True	True	True	True	False	True	True	True	True	True
True	False	False	False	False	True	True	False	True	False	True
True	False	False	False	True	True	False	True	True	False	True
True	False	False	True	False	True	True	True	False	False	True
True	False	False	True	True	True	False	True	False	False	True
True	False	True	False	False	False	True	False	True	False	True
True	False	True	False	True	False	True	True	True	False	True
True	False	True	True	False	False	True	True	False	False	True
True	False	True	True	True	False	True	True	False	False	True
True	True	False	False	False	True	True	False	True	True	True
True	True	False	False	True	True	False	True	True	True	True
True	True	False	True	False	True	True	True	True	True	True
True	True	False	True	True	True	False	True	True	True	True
True	True	True	False	False	False	True	False	True	True	True
True	True	True	False	True	False	True	True	True	True	True
True	True	True	True	False	False	True	True	True	True	True
True	True	True	True	True	False	True	True	True	True	True

That was a bit painful . . . let Python do the work . . . II

```
argument premise: (not s and c) and (not w or s) and (w or t) and (not t or h)
argument conclusion: h
```

c	h	s	t	w	not s and c	not w or s	w or t	not t or h	h	(not ((not s and c) and (not w or s) and (w or t) and (not t or h))) or (h)
False	False	False	False	False	False	True	False	True	False	True
False	False	False	False	True	False	False	True	True	False	True
False	False	False	True	False	False	True	True	False	False	True
False	False	False	True	True	False	False	True	False	False	True
False	False	True	False	False	False	True	False	True	False	True
False	False	True	False	True	False	True	True	True	False	True
False	False	True	True	False	False	True	True	False	False	True
False	False	True	True	True	False	True	True	False	False	True
False	True	False	False	False	False	True	False	True	True	True
False	True	False	False	True	False	False	True	True	True	True
False	True	False	True	False	False	True	True	True	True	True
False	True	False	True	True	False	False	True	True	True	True
False	True	True	False	False	False	True	False	True	True	True
False	True	True	False	True	False	True	True	True	True	True
False	True	True	True	False	False	True	True	True	True	True
False	True	True	True	True	False	True	True	True	True	True
True	False	False	False	False	True	True	False	True	False	True
True	False	False	False	True	True	False	True	True	False	True
True	False	False	True	False	True	True	True	False	False	True
True	False	False	True	True	True	False	True	False	False	True
True	False	True	False	False	False	True	False	True	False	True
True	False	True	False	True	False	True	True	True	False	True
True	False	True	True	False	False	True	True	False	False	True
True	False	True	True	True	False	True	True	False	False	True
True	True	False	False	False	True	True	False	True	True	True
True	True	False	False	True	True	False	True	True	True	True
True	True	False	True	False	True	True	True	True	True	True
True	True	False	True	True	True	False	True	True	True	True
True	True	True	False	False	False	True	False	True	True	True
True	True	True	False	True	False	True	True	True	True	True
True	True	True	True	False	False	True	True	True	True	True
True	True	True	True	True	False	True	True	True	True	True

All rows are True so we have a tautology