TUTORIAL 3 · [COMPUTATION AND LOGIC]

OBJECTIVES

In the first part of this tutorial, you will:

- work with *syllogisms* and meet Lewis Carroll, the logician.

In the second part you will:

- learn more about *sequents*, *satisfaction*, and *operations with predicates*;
- challenge the expressivity of *Aristotle's categorical propositions*.

TASKS

Exercises 1 and 4–6 are mandatory. Exercises 2, 3 and 7 are optional.

SUBMIT a file called `cl-tutorial-3` with your answers that do not require Haskell code (image or pdf) and the file `Things.hs` with your code.

DEADLINE Saturday, 10th of October, 4 PM UK time

Good Scholarly Practice

Please remember the good scholarly practice requirements of the University regarding work for credit.

You can find guidance at the School page

http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct.

This also has links to the relevant University pages. Please do not publish solutions to these exercises on the internet or elsewhere, to avoid others copying your solutions.

PART A

MANDATORY | BEFORE TUTORIAL SESSION

Read Chapters 7 (*Patterns of Reasoning*) and 8 (*More Patterns of Reasoning*) of the textbook.

Lewis Carroll, author of *Alice's Adventures in Wonderland* and *Through the Looking Glass*, taught logic at Oxford in the 1890s. In his words:

> *"When the terms of a proposition are represented by words, it is said to be in concrete form; when by letters, abstract."*

To translate a proposition from concrete to abstract form, we choose a universe and regard each term as a predicate, to which we assign a letter.

PART A (CONT.)

He gives the following example:

	All cats understand French.	$a \vDash b$
	Some chickens are cats.	$c \nvDash \neg a$
∴	Some chickens understand French.	$c \nvDash \neg b$

where *universe* = animals, a = cats, b = understanding French, c = chickens.

We can put this (sound) argument into inference form:

$$\frac{a \vDash b \qquad c \nvDash \neg a}{c \nvDash \neg b}$$

EXERCISE 1

MANDATORY | BEFORE TUTORIAL SESSION

Consider the following two arguments by Lewis Carroll:

All diligent students are successful.
All ignorant students are unsuccessful.
∴ Some diligent students are ignorant.

Every eagle can fly.
Some pigs cannot fly.
∴ Some pigs are not eagles.

Formulate them as syllogisms, then use Venn diagrams to check whether they are sound, i.e., whether they are correct arguments or not.

For each of the arguments, if it is sound, give a proof involving Venn diagrams, then derive it from Barbara using denying the conclusion, substituting for predicates, contraposition and the double negation law. If the argument is not sound, then give a counterexample.

SOLUTION TO EXERCISE 1

We start by formulating the first argument as a syllogism in symbolic form:

All diligent students are successful. $a \vDash b$

All ignorant students are unsuccessful. $c \vDash \neg b$

$\therefore$ Some diligent students are ignorant. $a \nvDash \neg c$

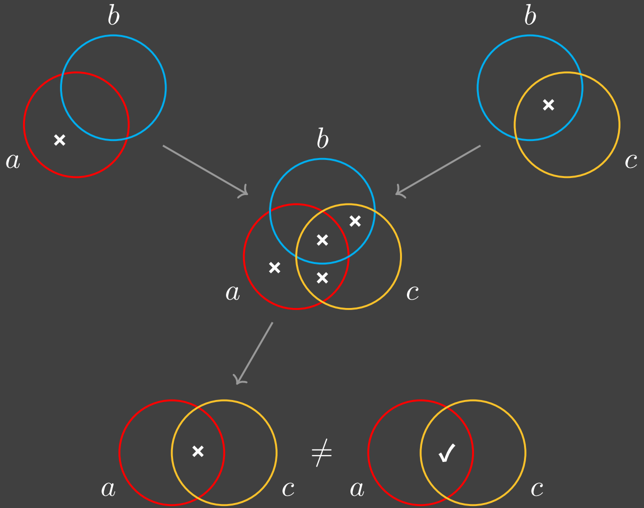
where a = diligent students, b = successful students, c = ignorant students.

$$\frac{a \vDash b \qquad c \vDash \neg b}{a \nvDash \neg c}$$

Then, we draw Venn diagrams to check whether it is sound.

SOLUTION TO EXERCISE 1 (CONT.)

We mark empty regions with × and inhabited regions with ✓.



SOLUTION TO EXERCISE 1 (CONT.)

From the assumptions, regions 4, 5, and 3, 7 of the diagram are empty. The conclusion requires regions 7 or 5 inhabited.

The argument is not sound: a counterexample is any universe with regions 3, 4, 5, 7 empty.

SOLUTION TO EXERCISE 1 (CONT.)

Now let us formulate the second argument as a syllogism in symbolic form:

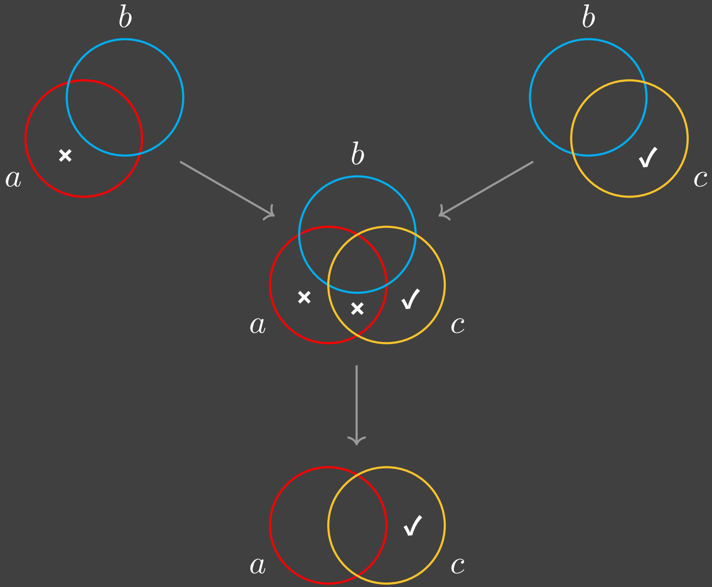
	Every eagle can fly.	$a \vDash b$
	Some pigs cannot fly.	$c \nvDash b$
$\therefore$	Some pigs are not eagles.	$c \nvDash a$

where a = eagles, b = things that can fly, c = pigs.

$$\frac{a \vDash b \qquad c \nvDash b}{c \nvDash a}$$

We now draw Venn diagrams to check whether it is sound.

SOLUTION TO EXERCISE 1 (CONT.)



SOLUTION TO EXERCISE 1 (CONT.)

From the assumptions, regions 4 and 5 of the diagram are empty, and 5 or 1 inhabited – therefore 1 is inhabited. The conclusion required 1 or 3 inhabited, therefore the argument is sound.

The argument is in the form Baroco, and we can derive it from Barbara by contraposition in two steps.

We start from Barbara:

$$\frac{c \vDash a \qquad a \vDash b}{c \vDash b}$$

by applying contraposition

we get Bocardo:

$$\frac{c \vDash a \qquad c \nvDash b}{a \nvDash b}$$

by applying contraposition

we get Baroco:

$$\frac{a \vDash b \qquad c \nvDash b}{c \nvDash a}$$

EXERCISE 2

OPTIONAL | BEFORE TUTORIAL SESSION

What can you say about the number of occurrences of $\nvDash$ in any sound syllogism? How about the number of occurrences of $\neg$?

Consider the following syllogism:

No animals are unicorns.
All unicorns are horses.
∴ Some horses are not animals.

Translate it to symbolic form and explain, using the answers given to the two questions above, without further calculation, and without giving a counterexample, why it is not sound.

SOLUTION TO EXERCISE 2

The number of occurences of $\nvDash$ in any sound syllogism, like the number of occurences of $\neg$ in any sound syllogism, is even.

This property is invariant when we substitute $\neg a$ for a predicate a and when we apply either kind of contraposition.

Let’s write our example in symbolic form:

	No animals are unicorns.	$a \vDash \neg b$
	All unicorns are horses.	$b \vDash c$
∴	Some horses are not animals.	$c \nvDash a$

where a = animals, b = unicorns, c = horses.

$$\frac{a \vDash \neg b \qquad b \vDash c}{c \nvDash a}$$

It is easy to see that the syllogism is not sound: we have an odd number of $\nvDash$ symbols (1), and an odd number of $\neg$ symbols (1).

EXERCISE 3

OPTIONAL | BEFORE TUTORIAL SESSION

In Chapter 8 of the textbook, at page 66, you saw the list of all valid syllogisms. In total, the list contains 15 syllogisms.

Can you convince yourself that this list is indeed exhaustive? That is, there is no other valid syllogism which is not on that list.

This is a challenging question: kudos if you solve it; if you don’t, no worries, it is safe to assume it is correct for the following exercise.

SOLUTION TO EXERCISE 3

One way of checking that there is no other sound syllogism besides the 15 ones in our list is to take each syllogism and check if it is sound. We'll give you below some hints on how to do this.

We know there are 256 syllogisms in total (you can read here why). Drawing Venn diagrams for all of them and checking their soundness would take a long time... What we can do instead is to eliminate from the list of all syllogisms some subsets of syllogisms that we know can't be sound.

To find those subsets of syllogisms that should be eliminated from the list, we should use Venn diagrams to identify patterns which lead to unsound syllogisms.

SOLUTION TO EXERCISE 3 (CONT.)

Here are some examples of such patterns of unsound syllogisms:

1. A sound syllogism can’t have two negative premises.
2. The conclusion of a sound syllogism must be negative, if either premise is negative.
3. No particular conclusion can be drawn from two universal premises.

Try to convince yourself that each of these three rules is correct by drawing Venn diagrams for a syllogism breaking the rule.

There are several such rules, called *Syllogistic Fallacies*. You can read more about these at https://en.wikipedia.org/wiki/Syllogistic_fallacy.

SOLUTION TO EXERCISE 3 (CONT.)

⌨ Another way to deal with this question is to use the computer to check for soundness all the 256 syllogisms.

The file `sound-syllogisms.hs` contains a Haskell representation of syllogisms and functions that help you translate them into Venn diagrams and check if a syllogism is sound.

We do not expect you to understand all this code now. But if you are interested, you can use it to deepen your understanding of syllogisms.

⌨ You can try to encode the syllogistic fallacies in the previous slide and add them to the file. Can you filter all the unsound syllogisms using your functions?

EXERCISE 3 (CONT.) 🏆

OPTIONAL | 🕒 BEFORE TUTORIAL SESSION

Aristotle considered 9 other syllogisms to be sound. This is not because he made a mistake, but because he made an assumption that we do not make: under the so-called *existential assumption*, $a \vDash b$ also requires that there is at least one thing that satisfies a.

To derive Aristotle's 9 extra correct arguments, it suffices to add the existential assumption as a rule with no premise of the form:

$$\frac{}{\text{“some } a \text{ is } a\text{”}}$$

✎ Convince yourself that the existential assumption can be expressed as “some a is a”. Then write the conclusion of the rule above as a sequent.

SOLUTION TO EXERCISE 3 (CONT.)

The existential assumption holds for our system of rules if and only if for every universal affirmative sequent $a \vDash b$, there is at least one thing that satisfies a.

This is equivalent to checking that for every universal affirmative sequent $a \vDash a$ (with a in place of b), there is at least one thing that satisfies a.

But $a \vDash a$ is trivially true. So all we need to enforce is that for every predicate a, there is a thing that satisfies a.

Intuitively, we can say “some a exists”. To formulate this as an Aristotelian proposition, we choose a statement that is trivially true for all elements that satisfy a: “some element that satisfies a satisfies a”, i.e., some a is a, or in symbols, $a \nvDash \neg a$.

EXERCISE 3 (CONT.) 🏆

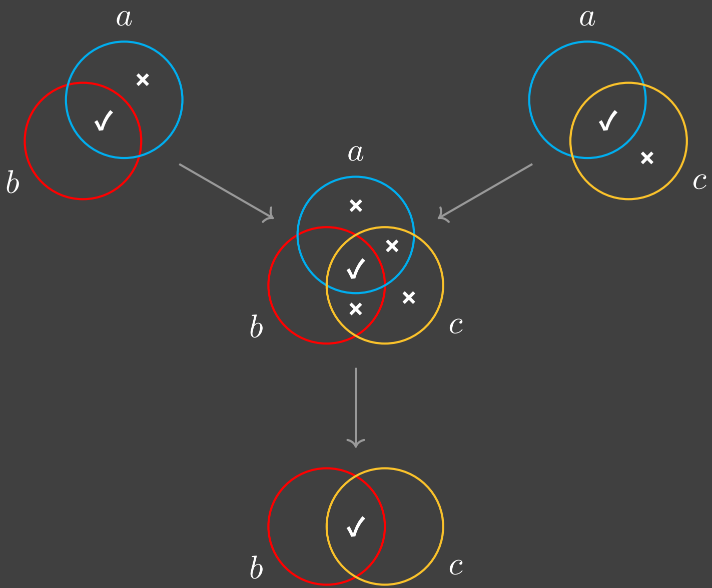
OPTIONAL | 🕒 BEFORE TUTORIAL SESSION

✏ Check that the following syllogism by Aristotle is correct under the existential assumption using Venn diagrams.

$$\frac{a \vDash b \qquad c \vDash a}{c \nvDash \neg b}$$

✏ Derive the syllogism from the 15 correct arguments in the textbook (page 66) and the *existential assumption rule* given on the previous slide.

SOLUTION TO EXERCISE 3 (CONT.)



SOLUTION TO EXERCISE 3 (CONT.)

Using the Venn diagrams on the previous slide we conclude that the syllogism is correct under the existential assumption.

The argument is in the form Barbari, and it can be derived from Darii, the existential assumption rule, and the substitution rule.

Let's recall Darii first, but with m, p, s instead of a, b and c in order to avoid confusion later on, when we apply Darii:

$$\frac{m \vDash p \qquad s \nvDash \neg m}{s \nvDash \neg p} \text{ (Darii)}$$

SOLUTION TO EXERCISE 3 (CONT.)

We derive the argument in three steps:

1. From the existential assumption (EA), we get

$$\frac{}{c \not\models \neg c}\ \text{(EA)}$$

2. Then, from Darii, substituting c for m, a for p, and c for s, we get:

$$\frac{c \models a \qquad c \not\models \neg c}{c \not\models \neg a}\ \text{(Darii)}$$

3. Using Darii again and substituting a for m, b for p, and c for s, we get:

$$\frac{a \models b \qquad c \not\models \neg a}{c \not\models \neg b}\ \text{(Darii)}$$

SOLUTION TO EXERCISE 3 (CONT.)

We put these three steps together and obtain the following derivation:

$$\dfrac{a \vDash b \qquad \dfrac{c \vDash a \qquad \dfrac{}{c \nvDash \neg c}}{c \nvDash \neg a}}{c \nvDash \neg b}$$

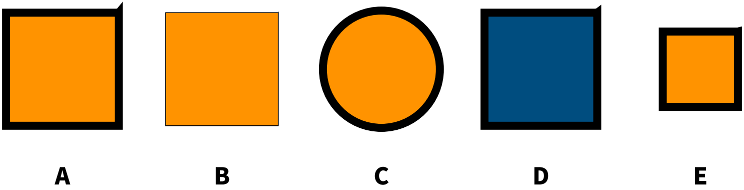
Note that the premises of this derivation are $a \vDash b$ and $c \vDash a$, while the conclusion is $c \nvDash \neg b$. Hence we have derived the argument:

$$\dfrac{a \vDash b \qquad c \vDash a}{c \nvDash \neg b}$$

PART B

MANDATORY | 🕒 BEFORE TUTORIAL SESSION

In this tutorial, we’ll be reusing the universe of 5 things from our previous tutorial:



⚠ The file `Things.hs` contains a description in Haskell of this universe. You will use it as a basis for your solutions to exercises that require Haskell code.

EXERCISE 4

MANDATORY | BEFORE TUTORIAL SESSION

Read the subsection on *Sequents* from Chapter 6 (*Features and Predicates*) of the textbook, on page 53.

Express each of the following using a sequent:

1. “Every big amber thing has a thick border” is false.
2. “Some small thing is a disc” is true.
3. “Some small square is amber” is false.

SOLUTION TO EXERCISE 4

1. “Every big amber thing has a thick border” is false.
   $\texttt{isBig}, \texttt{isAmber} \not\models \texttt{hasThickBorder}$

2. “Some small thing is a disc” is true.
   $\texttt{isSmall} \not\models \neg\,\texttt{isDisc}$

3. “Some small square is amber” is false.
   $\texttt{isSmall}, \texttt{isSquare} \models \neg\,\texttt{isAmber}$

EXERCISE 5

MANDATORY | BEFORE TUTORIAL SESSION

⌨ Define (and add to the file `Things.hs`) an infix function:

```
(|=) :: Predicate Thing -> Predicate Thing -> Bool
```

for testing whether a sequent involving one antecedent and one succedent is true or false. Use it to check that `isDisc ⊨ isAmber`.

⌨ Now implement an infix function:

```
(|/=) :: Predicate Thing -> Predicate Thing -> Bool
```

so that `a |/= b` is true if and only if some `a` is not `b`.

SOLUTION TO EXERCISE 5

```
(|=) :: Predicate Thing -> Predicate Thing -> Bool
a |= b = and [ b x | x <- things, a x ]

(|/=) :: Predicate Thing -> Predicate Thing -> Bool
a |/= b = or [ not (b x) | x <- things, a x ]
```

EXERCISE 5 (CONT.)

MANDATORY | BEFORE TUTORIAL SESSION

Define an infix function:

```
(||=) :: [Predicate Thing] -> Predicate Thing -> Bool
```

for testing whether a sequent involving a list of antecedents and one succedent is true or false. Use it to check that:

- isBlue, isSquare ⊨ isBig
- isBig, isAmber ⊭ isDisc

SOLUTION TO EXERCISE 5 (CONT.)

```
(||=) :: [Predicate Thing] -> Predicate Thing -> Bool
as ||= b = and [ b x | x <- things, and [a x | a <- as] ]

-- > [isBlue, isSquare] ||= isBig
-- True
-- > [isBig, isAmber] ||= isDisc
-- False
```

EXERCISE 6

MANDATORY | BEFORE TUTORIAL SESSION

Recall that the type `Predicate u` is defined as `u -> Bool`. The following function negates a predicate:

```
neg :: Predicate u -> Predicate u
(neg a) x = not (a x)
```

For example, `(neg isAmber) C = not (isAmber C) = False`, and `isBlue |= neg isSmall` produces `True`.

Define functions

```
(|:|) :: Predicate u -> Predicate u -> Predicate u
(&:&) :: Predicate u -> Predicate u -> Predicate u
```

that compute the disjunction and conjunction of two predicates.

SOLUTION TO EXERCISE 6

```haskell
(|:|) :: Predicate u -> Predicate u -> Predicate u
(a |:| b) x = a x || b x

(&:&) :: Predicate u -> Predicate u -> Predicate u
(a &:& b) x = a x && b x
```

EXERCISE 6 (CONT.)

MANDATORY | 🕒 BEFORE TUTORIAL SESSION

⌨ Which of the following produce `True`?

1. `isBig &:& isAmber |= isDisc`
2. `isBig &:& isDisc |= isAmber`
3. `isSmall &:& neg isBlue |= neg isDisc`
4. `isBig |:| isAmber |= neg isSquare`
5. `neg (isSquare |:| isBlue) |= hasThickBorder`
6. `neg isSquare &:& neg isAmber |= isDisc`

SOLUTION TO EXERCISE 6 (CONT.)

1.
   ```
   isBig &:& isAmber |= isDisc
    --> False
   ```

2.
   ```
   isBig &:& isDisc |= isAmber
    --> True
   ```

3.
   ```
   isSmall &:& neg isBlue |= neg isDisc
    --> True
   ```

4.
   ```
   isBig |:| isAmber |= neg isSquare
    --> False
   ```

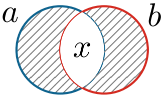
5.
   ```
   neg (isSquare |:| isBlue) |= hasThickContour
    --> True
   ```

6.
   ```
   neg isSquare &:& neg isAmber |= isDisc
    --> True
   ```

EXERCISE 7 🏆

Revisit section *Venn diagrams with inhabited regions* from Chapter 8 of the textbook, at page 64. We'll work now with diagrams whose regions are either empty (hatched) or inhabited (marked with an x, y, or z, etc.).

Given two predicates a and b, any Venn diagram with two circles (one for a, one for b) is uniquely determined by the set of *all* Aristotelian propositions ("Every a is b", "No a is b", "Some a is b", or "Some a is not b") that are valid for that diagram. For example, the Venn diagram



is determined by $\{$Every a is b, Every b is a, Some a is b, Some b is $a\}$. It's the only diagram with two circles for which these propositions are true.

Does this property still hold for three predicates (and Venn diagrams with three circles, one for each predicate)?

SOLUTION TO EXERCISE 7

For simplicity, we mark empty regions with × and inhabited regions with ✓.

Because Aristotelian propositions are binary, they can only talk about two regions at a time. They cannot refer to the intersection of a, b, and c. Therefore, using Aristotelian propositions we cannot distinguish between two Venn diagrams with the same marks in all regions except the middle region corresponding to the intersection of a, b, and c.