Crash Course on Logic and Set Theory

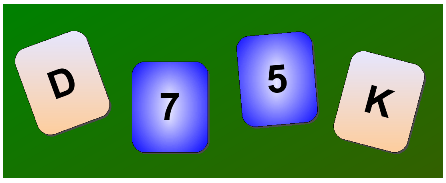
Blended Intensive Program: A Journey into Knowledge and Reasoning
22–23 June 2026

Pierre Talbot (pierre.talbot@uni.lu)

Goals

- ✯ Understanding the syntax and semantics of propositional logic.
- ✯ Study the usual set operations (union, intersection, inclusion, ...).
- ✯ Understanding the syntax and semantics of first-order logic.
- ✯ Modelling a simple game using first-order logic.

1 Wason Selection Task



source: Wikipedia

In psychology, the Wason selection task (or four-card problem) is a logic puzzle devised by Peter Cathcart Wason in 1966. It is one of the most famous tasks in the study of deductive reasoning. An example of the puzzle is:

> You are shown a set of four cards placed on a table, each of which has a number on one side and a letter on the other. The visible faces of the cards show D, 7, 5 and K. Which card(s) must you turn over in order to test that if a card shows a letter D on one face, then its opposite face is a 5?

A response that identifies a card that need not be inverted, or that fails to identify a card that needs to be inverted, is incorrect.

2 Syntax of Propositional Logic

Propositional logic is a mathematical formalism to express sentences and reason on them in a formal way. Propositional logic consists of *variables* $x, y, \ldots$ which are indivisible units of knowledge and *logical connectives* which combine them. The variables are denoted by a finite set $X = \{x, y, \ldots\}$. The set of *propositional formulas* is obtained with the following grammar where F and G are themselves propositional formulas:

$\langle F, G \rangle$ ::=

$\mid\ x$	*variable* $x \in X$
$\mid\ \neg F$	*negation*
$\mid\ F \wedge G$	*conjunction*
$\mid\ F \vee G$	*disjunction*
$\mid\ F \Rightarrow G$	*implication*
$\mid\ F \Leftrightarrow G$	*equivalence*
$\mid\ (\ F\ )$	*grouping*

For instance, $x \vee y \Rightarrow z$ is a valid logical formula denoting "x or y implies z". In a less abstract way, we could write "run or swim implies more tired", or in English "if I run or swim, then I'll be more tired". In this case, the variables represent the pieces of knowledge "run", "swim" and "more tired". The word "or" is formalized by the logical disjunction $\vee$, and the construction "if A then B" is formalized by the logical implication $\Rightarrow$. In the case of the implication, note that if the proposition A is false, then it does not say anything about B (it can hold or not).

Exercise 1 – From natural language to propositional logic

Translate the following statements into propositional formulas[1]. First isolate the pieces of knowledge of interest, then construct your formula with them. Note that logic does not decide what is true, it decides what makes sense with regards to a formal system.

1. If it is raining, then there are clouds in the sky.
2. To be or not to be.
3. You are shy and you are not shy.
4. (Association fallacy) They are racist, you voted for them, therefore you are racist.
5. (Circular Reasoning) I'm smart because I eat properly and I eat properly because I'm smart.
6. (Affirming the Consequent)

   > If my car was a Ferrari, it would be able to travel at over a hundred miles per hour. I clocked my car at 121 miles per hour. Therefore, my car is a Ferrari.

7. (Denying the Antecedent)

   > If a person is wearing a hat, they have a head. I am not wearing a hat. Therefore I do not have a head.

8. (Affirming a Disjunct)

   > I will eat cereal or drink coffee. I am going to eat cereal. Therefore, I cannot drink coffee.

end of exercise.

A **tautology** is a logical formula that is always true, regardless of whether its variable are true or not. Inversely, a **contradiction** is a logical formula that is always false.

Exercise 2 – Tautology and contradiction

Which formulas of the previous exercise are tautologies, and which ones are contradictions?

end of exercise.

Let us dive into a small technicality of propositional logic: **operator precedence**. In the sentence "if I run or swim, then I'll be more tired", the fact that we group "run" and "swim" together is due to context. However, what prevents us from understanding $x \vee y \Rightarrow z$ as $x \vee (y \Rightarrow z)$? To avoid ambiguity, we define precedence among operators: grouping, negation, conjunction, disjunction, implication, equivalence. Hence, $\neg x \wedge \neg(y \vee z \wedge z)$ is understood as $(\neg x) \wedge (\neg(y \vee (z \wedge z)))$ when fully parenthesized.

[1]Credits: some examples are taken from `https://tvtropes.org/pmwiki/pmwiki.php/UsefulNotes/LogicalFallacies`

Exercise 3 – Syntax of propositional logic

For the following formulas, state whether they are well-formed formula and if yes, give their fully parenthesized form.

- $x \wedge \neg y \vee z$
- $\neg\neg\neg y \vee z$
- $\neg x \neg y$
- $x \Rightarrow y \Leftrightarrow y \Rightarrow x$
- $x > y \Leftrightarrow x + 1 > y + 1$

end of exercise.

3 Logical Argument

An *argument* is one or more premises—sentences, statements, or propositions—directed towards arriving at a logical conclusion. A well-known argument is *modus ponens*, example:

- Premise 1 (**P1**): If it is raining, then there are clouds in the sky.
- Premise 2 (**P2**): It is raining.
- Conclusion (**C**): There are clouds in the sky.

An argument is formally written as $\mathbf{P1}, \mathbf{P2}, \ldots, \mathbf{Pn} \vdash \mathbf{C}$. This form is called a **sequent**. On the left of the turnstile $\vdash$, we have what we know (e.g. the facts, the knowledge base, ...), and on the right, the conclusion. Note that **C** is a valid conclusion only under the hypothesis that the premises $\mathbf{P1}, \mathbf{P2}$, ... are themselves true.

The modus ponens, in general, is written $A, A \rightarrow B \vdash B$; in English "Given that I know A, and that I know that if A then B; therefore I can conclude B."

Exercise 4 – Modus Ponens

Given the Wason selection test above, if we consider only the first card, write the argument allowing you to deduce that its opposite face must be 5.

end of exercise.

Another well-known argument is *modus tollens*, example:

- If Rex is a chicken, then he is a bird.
- Rex is not a bird.
- Therefore, Rex is not a chicken.

Exercise 5 – Modus Tollens

- Give the sequent of modus tollens.
- Given the Wason selection test above, if we consider only the second card, write the argument allowing you to deduce that its opposite face must not be D.

end of exercise.

An argument is **valid** iff whenever its premises are true, the conclusion must also be true. An argument is **sound** iff it is valid and all its premises are true.

4 Sequent Calculus

If I give you a logical formula, such as modus ponens: $(A \wedge (A \Rightarrow B)) \Rightarrow B$, how can you show it is a tautology? We have supposed it is a valid argument, but how to prove it? This is the job of **proof system**, and we are going to (briefly) study a very elegant one called **sequent calculus**.

A sequent is a pair between premises and conclusions $A \vdash B$. In the sequent calculus, the premises are conjunctively connected, and the conclusions are disjunctively connected. In other words, a full-fledged sequent is of the form $A_1, \ldots, A_n \vdash B_1, \ldots, B_k$, and must be understood as "Given the premises $A_1 \wedge \ldots \wedge A_n$, we can deduce one of the conclusions, written $B_1 \vee \ldots \vee B_k$.".

The following **inference rules** form a system to formally prove whether a formula is a tautology or not. Let Γ, Δ be sequences of formulas, e.g. $\Gamma = a \vee b, \neg b, b \wedge c$. An inference rule is read bottom-up: to prove the sequent at the bottom, one must prove the sequent(s) above the inference line.

$$\dfrac{}{\Gamma, A \vdash A, \Delta}\ \textsc{Axiom} \qquad \dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \textsc{Left-}\wedge \qquad \dfrac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ \textsc{Right-}\wedge \qquad \dfrac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta}\ \textsc{Left-}\Rightarrow$$

$$\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta}\ \textsc{Right-}\Rightarrow \qquad \dfrac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ \textsc{Left-}\neg \qquad \dfrac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ \textsc{Right-}\neg$$

For instance, to prove modus ponens is a tautology, written $\vdash (A \wedge (A \Rightarrow B)) \Rightarrow B$, we can use the following **derivation tree**:

$$\dfrac{\dfrac{\dfrac{\overset{\textsc{Axiom}}{A \vdash A, B} \qquad \overset{\textsc{Axiom}}{A, B \vdash B}}{A, A \Rightarrow B \vdash B}\ \textsc{Left-}\Rightarrow}{A \wedge (A \Rightarrow B) \vdash B}\ \textsc{Left-}\wedge}{\vdash (A \wedge (A \Rightarrow B)) \Rightarrow B}\ \textsc{Right-}\Rightarrow$$

Exercise 6 – Proving modus tollens

Build the derivation tree of the sequent for modus tollens.

end of exercise.

Exercise 7 – Affirming the consequent

Build the derivation tree of the sequent $\vdash (B \wedge A \Rightarrow B) \Rightarrow A$. Note that this logical argument is flawed, so you should be stuck in the derivation tree, with no rule to apply; showing it cannot be proven.

end of exercise.

Exercise 8 – Rewriting Implication into Disjunction (bonus)

After looking for the sequent rules for disjunction, prove that $(A \Rightarrow B) \Leftrightarrow (\neg A \vee B)$ is a tautology. Note that $A \Leftrightarrow B$ is equivalent to $A \Rightarrow B \wedge B \Rightarrow A$.

end of exercise.

5 Semantics of Propositional Logic

The semantics of propositional logic consists in assigning **truth values** to the propositional variables of a given formula. We denote by $\mathbf{T}$ the value "true", and by $\mathbf{F}$ the value "false".

The **truth tables** of the negation and conjunction are as follows:

F	$\neg F$
$\mathbf{T}$	$\mathbf{F}$
$\mathbf{F}$	$\mathbf{T}$

F	G	$F \wedge G$
$\mathbf{T}$	$\mathbf{T}$	$\mathbf{T}$
$\mathbf{F}$	$\mathbf{T}$	$\mathbf{F}$
$\mathbf{T}$	$\mathbf{F}$	$\mathbf{F}$
$\mathbf{F}$	$\mathbf{F}$	$\mathbf{F}$

Exercise 9 – Truth Tables

Write the truth tables for disjunction, implication and equivalence.

end of exercise.

Given a formula $(x \vee \neg y)$, a possible assignment making the formula true is $x \mapsto \mathbf{T}, y \mapsto \mathbf{F}$, but it is not the only such assignment. Indeed, $x \mapsto \mathbf{T}, y \mapsto \mathbf{T}$ also makes the formula true. Since there exists an assignment making the formula true, it is not a contradiction. However, it is not a tautology neither, because we can find an assignment making the formula false: $x \mapsto \mathbf{F}, y \mapsto \mathbf{T}$.

Let asn be an assignment, a function from formulas to truth values. Then the truth value of a formula F under assignment asn is given by the following **semantic consequence** $\vDash$:

$$\begin{array}{ll} \vDash_{asn} x & \text{iff } asn(x) = \mathbf{T} \\ \vDash_{asn} F_1 \wedge F_2 & \text{iff } \vDash_{asn} F_1 \text{ and } \vDash_{asn} F_2 \\ \vDash_{asn} F_1 \vee F_2 & \text{iff } \vDash_{asn} F_1 \text{ or } \vDash_{asn} F_2 \\ \vDash_{asn} F_1 \Rightarrow F_2 & \text{iff } \vDash_{asn} F_1 \text{ implies } \vDash_{asn} F_2 \\ \vDash_{asn} \neg F & \text{iff } \vDash_{asn} F \text{ does not hold} \end{array}$$

Theorem 1. *Propositional logic is sound and complete. For all formulas F:*

- *Soundness: if $\vdash F$, then $\vDash_{asn} F$ for all assignments asn.*
- *Completeness: if $\vDash_{asn} F$ for all assignments asn, then $\vdash F$.*

Exercise 10 – Assignments

- Let $asn = \{x \mapsto \mathbf{T}, y \mapsto \mathbf{F}, z \mapsto \mathbf{T}\}$, verify that $\vDash_{asn} (x \wedge y) \vee (y \wedge z) \vee (x \wedge z)$.

- Find an assignment asn such that $\vDash_{asn} x \Rightarrow (\neg y \vee \neg x)$ holds.

end of exercise.

6 Connection to this BIP

- The **satisfiability problem** (SAT) consists in finding an assignment asn given a formula F such that $\vDash_{asn} F$. A **SAT solver** is a computer program trying to solve this task as quickly as possible. However, as the satisfiability problem is NP-complete, it is a challenging task.
- **Constraint programming** is an extension of SAT allowing functions and predicates, e.g. $x + 1 = y \wedge y = 2$ (see below first-order logic).
- **Logic programming** can be viewed as turning modus ponens (deductive reasoning) into a programming language with constructions of the form $A \Leftarrow B$ (To deduce A, I need to deduce B first).
- **Rewriting systems** enables to rewrite logical formula into (equivalent) formulas, in order to simplify them, e.g. $(a \wedge b) \vee (\neg a \wedge b) \Leftrightarrow b$.
- **Argumentation** is another logical paradigm, which supports contradictions in the knowledge base.
- **Neural network verification**: turn a neural network with some required pre/post-conditions into a logical formula and verify its validity.

7 Set Theory

We follow the course from Stanford available here: `https://web.stanford.edu/class/archive/cs/cs103/cs103.1254/lectures/00/Lecture%20Slides.pdf`.

Exercise 11
Decide which of the following statements are true:

- $\{a, b, c\} = \{c, a, b\}$
- $\{1, ab\} = \{a, b, 1\}$
- $\{1\} \in \{1, 2, 3\}$
- $\{1\} \in \{\{1\}, \{2\}, \{1, 2\}\}$
- $2 \in \{1, 2\} \in \{\{1\}, \{2\}, \{1, 2\}\}$

Exercise 12
Give the set generated by $\{x \in \mathbb{N} | x > 0 \wedge x < 5\}$.

Exercise 13
What is the result of $(\{1, 2, 3\} \cup \{1, 4\}) \cap (\{4, 2\} \cup \{1\})$?

Exercise 14
Decide which of the following statements are true:

- $\{a, b\} \subseteq \{c, a, b\} \cap \{1, 2\}$
- $\{\} \subseteq \{1, 2\} \cap \{1, 2\}$
- $\{1\} \subseteq \{\{1\}, \{2\}, \{1, 2\}\}$
- $\{\} \subseteq \{1, 2\}$
- $\{\} \subseteq \{\{1\}, \{2\}, \{1, 2\}\}$
- $\{\{\}\} \subseteq \{\{1\}, \{2\}, \{1, 2\}\}$
- $\{\{1\}\} \subseteq \{\{1\}, \{2\}, \{1, 2\}\}$
- $\{2, 3\} \in \mathcal{P}(\{1, 2, 3\})$
- $\{\{\}, \{a\}, \{b\}\} \subseteq \mathcal{P}(\{a, b, c\})$

Exercise 15
Decide which of the following statements are true for any set A, B, C:

- $A \subseteq B \Leftrightarrow A \in \mathcal{P}(B)$
- $\{1\} \subseteq \{1, 2\} \Leftrightarrow \{1\} \in \{\{\}, \{1\}, \{2\}, \{1, 2\}\}$
- $A \subseteq A \cup B$
- $A \subseteq A \cap B$
- $A \cap B \subseteq A \cup C$
- $A \cup (B \cup C) = (A \cup B) \cup C$
- $A \cup (B \cap C) = (A \cup B) \cap C$
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

8 Motivation for First-order Logic

It is not always easy (or possible) to turn a statement into a propositional logic formula. Take for example:

> "all humans are mortals, Socrates is human, therefore Socrates is mortal"

A propositional formula could look like:

> "((human ⇒ mortal) ∧ Socrates-human) ⇒ Socrates-mortal"

From this propositional formula, if we know Socrates is human, we cannot deduce Socrates is mortal. Arguably, we could use a more general formulation such as:

> "((human ⇒ mortal) ∧ human) ⇒ mortal"

but this does not make the difference between two humans: we lose the knowledge that Socrates is human. Another possibility is that we know the list of all humans, for the sake of example, let us suppose it is "Ada, Socrates, Grace". We could create an implication formula for each distinct human:

> (ada-human ⇒ ada-mortal) ∧ (socrates-human ⇒ socrates-mortal) ∧ (grace-human ⇒ grace-mortal)

From there, we can use Modus Ponens whenever we know "ada-human" or "socrates-human" or "grace-human".

The formula grows according to the number of humans under consideration. More problematic is the case where the set on which we are reasoning about is infinite, take for instance numbers. Take the following statement:

> "If x is an integer, then x+1 is greater than x"

It becomes impossible to write the corresponding propositional formula, which could look like the infinite formula:

> (x_is_0 ⇒ x_is_1_>_x_is_0) ∧ (x_is_1 ⇒ x_is_2_>_x_is_1) ∧ . . .

Therefore, we need a new construction in our logic to reason over collection of things, and be able to say things such as "for all humans, ...". This extension is called first-order logic.

9 Syntax of First-order Logic

First-order logic (FOL) extends propositional logic with quantifiers, functions and predicates. In FOL, the previous sentence is written:

Premise 1	$(\forall x,\ human(x) \Rightarrow mortal(x))$	(all humans are mortals)
Premise 2	$human(Socrates)$	(Socrates is human)
Conclusion	$mortal(Socrates)$	(Socrates is mortal)

The universal quantifier $\forall x, F$ is read "for all variables x, the formula F holds". The formula also have two predicates $human$ and $mortal$, and one constant $Socrates$ (a constant is a function of arity 0). A predicate $P(T_1, \ldots, T_n)$ is a relation which is true for some combinations of its arguments $T_1, \ldots, T_n$, and false for others.

The formal syntax of FOL is given as follows. Let $S = \langle X, F, P \rangle$ be a *first-order signature* where X set of variables, F set of function symbols and P set of predicate symbols.

$\langle T_1, \ldots, T_n \rangle ::= x$ — *variable* $x \in X$
$\mid\ f(T_1, \ldots, T_n)$ — *function* $f \in F$

$\langle F, G \rangle ::=$
$\mid\ \neg F$ — *negation*
$\mid\ F \wedge G$ — *conjunction*
$\mid\ F \vee G$ — *disjunction*
$\mid\ F \Rightarrow G$ — *implication*
$\mid\ F \Leftrightarrow G$ — *equivalence*
$\mid\ p(T_1, \ldots, T_n)$ — *predicate* $p \in P$
$\mid\ \exists x,\ F$ — *existential quantifier*
$\mid\ \forall x,\ F$ — *universal quantifier*

Let Φ the set of well-formed formulas.

Exercise 16 – From natural language to FOL

- Everything is bitter or sweet. $\forall x,\ bitter(x) \vee sweet(x)$.
- Either everything is bitter or everything is sweet. $(\forall x,\ bitter(x)) \vee (\forall x,\ sweet(x))$
- There is somebody who is loved by everyone. $\exists x,\ \forall y,\ love(y, x)$
- Nobody is loved by no one.
- If someone is noisy, everybody is annoyed.
- Frogs are green.
- Frogs are not green.
- No frog is green.
- Some frogs are not green.
- A mechanic likes Bob.
- A mechanic likes herself.
- Every mechanic likes Bob.
- Some mechanic likes every nurse.

- There is a mechanic who is liked by every nurse.

Taken from https://disi.unitn.it/~bernardi/Courses/LSNL/Slides/fl1.pdf.
end of exercise.

- $\neg\exists x, \neg\forall y, loved(x, y)$
- $\forall x, \neg\neg\forall y, loved(x, y)$
- $\forall x, \forall y, loved(x, y)$

To push logical negation inside the quantifiers, we have the following two equivalences:

- $\neg\forall x, F \Leftrightarrow \exists x,\ \neg F$
- $\neg\exists x, F \Leftrightarrow \forall x,\ \neg F$

It can be used to simplify a formula.

Note that sequent calculus for propositional logic is easily extended to FOL with the addition of 4 rules for the two quantifiers (left and right versions). It could be used to prove the equivalences above. Furthermore, it is interesting to know that FOL is also a sound and complete logic.

Exercise 17 – A modelling exercise

Gameplay from Wikipedia (https://en.wikipedia.org/wiki/Dobble):

> The game is played with a deck of 55 cards, each featuring eight different symbols. Any two cards share exactly one matching symbol. The objective is to be the first player to identify and announce the common symbol between two cards.

Write a FOL formula to verify that each pair of cards has only one symbol in common. You can use integers to number the cards from 1 to 55, and the symbols from 1 to 57. You can use a predicate $card(i, j)$ which holds whenever the card i has the symbol j on it.

10 Semantics of First-order Logic

A *structure* A is a tuple $(U, [\![]\!]_F, [\![]\!]_P)$ where

1. U is a non-empty set of elements—called the *universe of discourse*,

2. $[\![]\!]_F$ is a function mapping function symbols $f \in F$ with arity n to interpreted functions $[\![f]\!]_F : U^n \to U$, and
3. $[\![]\!]_P$ is a function mapping predicate symbols $p \in P$ with arity n to interpreted predicates $[\![p]\!]_P \subseteq U^n$.

An *assignment* is a function $X \to U$ mapping variables to values. We denote the set of assignments by $\mathbf{Asn}$. Let $asn \in \mathbf{Asn}$, we write $asn[x \mapsto d]$ the assignment in which we updated the value of x by d in asn.

Examples. A common universe of discourse is the set of integers $\mathbb{Z}$. It has arithmetic functions $F = \{+, -, *, \ldots\}$ and predicates $P = \{<, \leq, =, \neq, \ldots\}$. We consider the standard interpretation of those function and predicate symbols, e.g.

- $[\![+]\!]_F = f \qquad$ where $f(x, y) \triangleq x + y$
- $[\![=]\!]_P = \{(0,0), (1,1), (-1,-1), (2,2), \ldots\} = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} \mid x = y\}$
- $[\![<]\!]_P = \{(0,1), (-1,0), (0,2), (-2,0), \ldots\} = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} \mid x < y\}$

The syntax and semantics are related by the ternary relation $A \vDash_{asn} \varphi$, called the *entailment*, where A is a structure, $asn \in \mathbf{Asn}$ and $\varphi \in \Phi$. It is read as "the formula φ is satisfied by the assignment asn in the structure A". We first give the interpretation function $[\![]\!]_{asn}$ for evaluating the terms of the language:

$$\begin{array}{ll} [\![x]\!]_{asn} & = asn(x) \text{ if } x \in X \\ [\![f(t_1, \ldots, t_n)]\!]_{asn} & = [\![f]\!]_F([\![t_1]\!]_{asn}, \ldots, [\![t_n]\!]_{asn}) \end{array}$$

The relation $\vDash$ is defined inductively as follows:

$$\begin{array}{ll} A \vDash_{asn} p(t_1, \ldots, t_n) & \text{iff } ([\![t_1]\!]_{asn}, \ldots, [\![t_n]\!]_{asn}) \in [\![p]\!]_P \\ A \vDash_{asn} \varphi_1 \wedge \varphi_2 & \text{iff } A \vDash_{asn} \varphi_1 \text{ and } A \vDash_{asn} \varphi_2 \\ A \vDash_{asn} \varphi_1 \vee \varphi_2 & \text{iff } A \vDash_{asn} \varphi_1 \text{ or } A \vDash_{asn} \varphi_2 \\ A \vDash_{asn} \neg\varphi & \text{iff } A \vDash_{asn} \varphi \text{ does not hold} \\ A \vDash_{asn} \exists x,\ \varphi & \text{iff there exists } d \in U \text{ such that } A \vDash_{asn[x \mapsto d]} \varphi \\ A \vDash_{asn} \forall x,\ \varphi & \text{iff for all } d \in U, \text{ we have } A \vDash_{asn[x \mapsto d]} \varphi \end{array}$$

Exercise 18 – Finding assignments

Consider the structure $A = \langle \{0, 1\}, [\![]\!]_F, [\![]\!]_P \rangle$ with $F = \{\}$ and $P = \{P, R\}$. The interpretation of the predicates is as follows: $[\![P]\!]_P = \{0, 1\}$ and $[\![R]\!]_P = \{(0, 0), (0, 1)\}$. Verify whether the following formulas are true in A:

- $\forall x,\ P(x)$
- $P(0)$
- $\neg R(0, 0)$
- $\exists x,\ R(x, x)$
- $\forall x,\ R(x, x)$
- $\forall x,\ R(x, x) \Rightarrow P(x)$
- $\forall x,\ \neg R(x, x) \Rightarrow P(x)$
- $\forall x,\ P(x) \Rightarrow \neg R(x, x)$

Taken from `https://disi.unitn.it/~bernardi/Courses/LSNL/Slides/fl1.pdf`.
end of exercise.

Note that when the universe of discourse U is finite, we can always turn a FOL formula into a propositional formula. This allows to use a SAT solver to solve formulas using more expressive symbols. The PICAT solver is a solver translating FOL formula (with a finite universe of discourse) to propositional formulas. OR-Tools or Choco are constraint programming solvers directly working on the FOL formula (without quantifiers).

11 FOL and Sets

How can we prove the validity of a statement such as $A \subseteq B$? We can rely on the first-order logic formula $\forall x,\ x \in A \Rightarrow x \in B$. Using this formula, we can attempt to show that whenever x is in A, then it is necessarily in B.

Example. Take for instance $A = \{x \in \mathbb{Z} \mid x > 10\}$ and $B = \mathbb{Z}$. To prove $A \subseteq B$, we can say:

$$\begin{array}{ll} A \subseteq B & \\ \Leftrightarrow \forall x,\ x \in A \Rightarrow x \in B & \text{definition of } \subseteq \\ \Leftrightarrow \forall x,\ x \in \{y \in \mathbb{Z} \mid y > 10\} \Rightarrow x \in \mathbb{Z} & \text{definition of } A \text{ and } B \\ \Leftrightarrow \forall x,\ (x \in \mathbb{Z} \wedge x > 10) \Rightarrow x \in \mathbb{Z} & \text{definition of } \in \\ \Leftrightarrow \forall x,\ x \in \mathbb{Z} \Rightarrow x \in \mathbb{Z} & \text{tautology} \end{array}$$

Exercise 19 – FOL definition of set predicates
Using a FOL formula, give the definition of the following set predicates:

- $A = B$
- $A = B \cup C$ (definition of union)
- $A = B \cap C$ (definition of intersection)
- $A \subset B$ where $\subset$ means the strict inclusion.

end of exercise.

Let A be a structure. Given a FOL formula φ, we can describe the set of all the assignments asn satisfying $A \vDash_{asn} \varphi$ as follows:

$$\begin{array}{l} sol : \Phi \to \mathcal{P}(\mathbf{Asn}) \\ sol(\varphi) = \{asn \in \mathbf{Asn} \mid A \vDash_{asn} \varphi\} \end{array}$$

Exercise 20 – Connecting FOL and set theory
We explore a connection between the logical connectives and the set operations:

- $sol(F \wedge G) = sol(F) \cap sol(G)$
- $sol(F \vee G) =$
- $sol(\neg F) =$
- $sol(F \Rightarrow G) =$

end of exercise.

12 Solutions

Exercise 1

1. $raining \Rightarrow clouds$
2. $be \vee \neg be$
3. $shy \wedge \neg shy$
4. $(themracists \wedge votedforthem) \Rightarrow youracist$
   In the following answers, we abstract the "unit of knowledge" with letters (but you can name them as you want).

5. $(A \Rightarrow B) \wedge (B \Rightarrow A)$
6. $((A \Rightarrow B) \wedge B) \Rightarrow A$
7. $((A \Rightarrow B) \wedge \neg A) \Rightarrow \neg B$
8. $(((A \vee B) \wedge A) \Rightarrow \neg B$

Exercise 2

(2) is a tautology, (3) is a contradiction.

Exercise 3

- $((x \wedge (\neg y)) \vee z)$
- $((\neg(\neg(\neg y))) \vee z)$
- $\neg x \neg y$: not well-formed!
- $((x \Rightarrow y) \Leftrightarrow (y \Rightarrow x))$
- $x > y \Leftrightarrow x + 1 > y + 1$: not a propositional logic formula.

Exercise 4

- Propositional formula: $((D \Rightarrow 5) \wedge D) \Rightarrow 5$
- Sequent: $(D \Rightarrow 5), D \vdash 5$
- (both are equivalent w.r.t. sequent calculus presented below).

Exercise 5

- Sequent of Modus Tollens: $(A \Rightarrow B), \neg B \vdash \neg A$
- Sequent applied to Wason test: $(D \Rightarrow 5), \neg 5 \vdash \neg D$

Exercise 6

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{\text{AXIOM}}{A \vdash A, B} \qquad \overset{\text{AXIOM}}{A, B \vdash B}}{A, A \Rightarrow B \vdash B}\ \text{LEFT-}\Rightarrow}{A \Rightarrow B \vdash \neg A, B}\ \text{RIGHT-}\neg}{\neg B, A \Rightarrow B \vdash \neg A}\ \text{LEFT-}\neg}{\neg B \wedge (A \Rightarrow B)) \vdash \neg A}\ \text{LEFT-}\wedge}{\vdash (\neg B \wedge (A \Rightarrow B)) \Rightarrow \neg A}\ \text{RIGHT-}\Rightarrow$$

Exercise 9

F	G	$F \vee G$
T	**T**	**T**
F	**T**	**T**
T	**F**	**T**
F	**F**	**F**

F	G	$F \Rightarrow G$
$\mathbf{T}$	$\mathbf{T}$	$\mathbf{T}$
$\mathbf{F}$	$\mathbf{T}$	$\mathbf{T}$
$\mathbf{T}$	$\mathbf{F}$	$\mathbf{F}$
$\mathbf{F}$	$\mathbf{F}$	$\mathbf{T}$

F	G	$F \Leftrightarrow G$
$\mathbf{T}$	$\mathbf{T}$	$\mathbf{T}$
$\mathbf{F}$	$\mathbf{T}$	$\mathbf{F}$
$\mathbf{T}$	$\mathbf{F}$	$\mathbf{F}$
$\mathbf{F}$	$\mathbf{F}$	$\mathbf{T}$

Exercise 10

- $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (x \wedge y) \vee (y \wedge z) \vee (x \wedge z)$
- $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (x \wedge y)$ or $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (y \wedge z)$ or $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (x \wedge z)$
- ($\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} x$ and $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} y$) or $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (y \wedge z)$ or $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (x \wedge z)$
- (true and false) or $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (y \wedge z)$ or $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (x \wedge z)$
- false or false or $\models_{\{x\mapsto\mathbf{T},y\mapsto\mathbf{F},z\mapsto\mathbf{T}\}} (x \wedge z)$
- false or false or true
- true

An assignment asn satisfying $\models_{asn} x \Rightarrow (\neg y \vee \neg x)$ is for instance $\{x \mapsto \mathbf{F}, y \mapsto \mathbf{T}\}$, or $\{x \mapsto \mathbf{T}, y \mapsto \mathbf{F}\}$

Exercise 11

- $\{a, b, c\} = \{c, a, b\}$: true
- $\{1, ab\} = \{a, b, 1\}$: false
- $\{1\} \in \{1, 2, 3\}$: false
- $\{1\} \in \{\{1\}, \{2\}, \{1, 2\}\}$: true
- $2 \in \{1, 2\} \in \{\{1\}, \{2\}, \{1, 2\}\}$: true

Exercise 12

$\{1, 2, 3, 4\}$

Exercise 13

$\{1, 2, 4\}$

Exercise 14

- $\{a, b\} \subseteq \{c, a, b\} \cap \{1, 2\}$: false
- $\{\} \subseteq \{1, 2\} \cap \{1, 2\}$: true
- $\{1\} \subseteq \{\{1\}, \{2\}, \{1, 2\}\}$: false
- $\{\} \subseteq \{1, 2\}$: true
- $\{\} \subseteq \{\{1\}, \{2\}, \{1, 2\}\}$: true
- $\{\{\}\} \subseteq \{\{1\}, \{2\}, \{1, 2\}\}$: false
- $\{\{1\}\} \subseteq \{\{1\}, \{2\}, \{1, 2\}\}$: true
- $\{2, 3\} \in \mathcal{P}(\{1, 2, 3\})$: true
- $\{\{\}, \{a\}, \{b\}\} \subseteq \mathcal{P}(\{a, b, c\})$: true

Exercise 15

- $A \subseteq B \Leftrightarrow A \in \mathcal{P}(B)$: true
- $\{1\} \subseteq \{1, 2\} \Leftrightarrow \{1\} \in \{\{\}, \{1\}, \{2\}, \{1, 2\}\}$: true
- $A \subseteq A \cup B$: true
- $A \subseteq A \cap B$: false
- $A \cap B \subseteq A \cup C$: true
- $A \cup (B \cup C) = (A \cup B) \cup C$: true
- $A \cup (B \cap C) = (A \cup B) \cap C$: false
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$: true
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$: true

Exercise 16

Solutions in `https://disi.unitn.it/~bernardi/Courses/LSNL/Slides/fl1.pdf`.

Exercise 17

$\forall c_1, c_2,\ (c_1 \neq c_2 \wedge c_1 \in \{1, \ldots, 55\} \wedge c_2 \in \{1, \ldots, 55\}) \Rightarrow \exists! s,\ s \in \{1, \ldots, 57\} \Rightarrow \mathit{card}(c_1, s) \wedge \mathit{card}(c_2, s)$
"For all pairs of distinct cards c_1 and c_2, there exists a unique symbol s appearing on both cards."

The symbol $\exists! x,\ P$ means "there exists an unique x for which P holds". It can be defined as follows:

$$\exists x,\ (P \wedge \forall y,\ x = y \vee \neg P)$$

Exercise 18

Solutions in `https://disi.unitn.it/~bernardi/Courses/LSNL/Slides/fl1.pdf`.

Exercise 19

- $A = B \Leftrightarrow (\forall x,\ x \in A \Leftrightarrow x \in B)$
- $A = B \cup C \Leftrightarrow (\forall x,\ x \in A \Leftrightarrow (x \in B \vee x \in C)$
- $A = B \cap C \Leftrightarrow (\forall x,\ x \in A \Leftrightarrow (x \in B \wedge x \in C))$
- $A \subset B \Leftrightarrow (\forall x,\ x \in A \Rightarrow x \in B) \wedge (\exists x,\ \neg(x \in A \wedge x \in B))$

Exercise 20

- $sol(F \wedge G) = sol(F) \cap sol(G)$
- $sol(F \vee G) = sol(F) \cup sol(G)$
- $sol(\neg F) = U \setminus sol(F)$ where U is the universe of discourse.
- $sol(F \Rightarrow G) = sol(F \wedge G) \cup sol(\neg F) = sol(\neg F \vee G)$