CHAPTER 4

General Proof Systems: Syntax and Semantics

Proof systems are built to prove, construct formal proofs of statements formulated in a given language formulated in a given language. First component of any proof system is hence its formal language $\mathcal{L}$. Proof systems can be thought as an *inference machine* with special statements, called *provable statements*, or *theorems* being its final products. The starting points are called *axioms* of the proof system. We distinguish two kinds of axioms: *logic* LA and *specific* SA.

When building a proof system for a given language and its semantics i.e. for a logic defined semantically we choose as a set of logical axioms LA some subset of tautologies, i.e. statements always true. This is why we call them logical axioms. A proof system with only logic axioms LA is also called *logic* proof systems, or just *proof systems* for short. If we build a proof system for which there is no known semantics, as it has happened in the case of classical, intuitionistic, and modal logics, we think about the logical axioms as statements universally true. We choose as axioms (finite set) the statements we for sure want to be universally true, and whatever semantics follows they must be tautologies with respect to it. Logical axioms are hence not only tautologies under an established semantics, but they also guide us how to establish a semantics, when it is yet unknown.

For the set of specific axioms SA we choose these formulas of the language that describe our knowledge of a universe we want to prove facts about. They are not universally true, they are true only in the universe we are interested to describe and investigate. This is why we call them specific axioms. A proof system with logical axioms LA and specific axioms SA is called *a formal theory* based on a proof system with logic axioms LA.

The inference machine is defined by a finite set of rules, called *inference rules*. The inference rules describe the way we are allowed to transform the information within the system with axioms as a staring point. The process of this transformation is called *a formal proof*. The provable formulas for which we have a formal proof are called consequences of the axioms, or theorem, or just simple *provable formulas*. We use proof systems not only to be able to build formal proofs in them, but also to search for proofs of given statements of their the language. We distinguish special proof systems for which it is possible to define a mechanical method for determining, given any statement of A, but which also generates a proof, is called *syntactically decidable* or *automatically decidable*, or *an automated system*

When building a proof system we choose not only axioms of the system, but also specific rules of inference. The choice of rules is often linked, as was the choice of axioms, with a given semantics. We want the rules to preserve the truthfulness of what we are proving from axioms via the rules. Rules with this property are called *sound rules* and the system *a sound proof system.* The notion of truthfulness is always defined by a given propositional, or predicate language $\mathcal{L}$ semantics **M**. Rules of inference can be sound under one semantics and not sound under another. When developing a proof system S the first goal is prove a theorem, called *Soundness Theorem* about its relationship with its semantics **M**. It states that the following holds for any formula A of the language $\mathcal{L}$ of the system S. *If a formula A is provable from logical axioms LA of S only, then A is a tautology under the semantics* **M**.

A proof system can be sound under one semantics, and not sound under the other. For example a set of axioms and rules sound under classical logic semantics might not be sound under intuitionistic semantics, **H**, **L**, **K** semantics, or others. This is why we talk about proof systems for classical logic, intuitionistic logic, for modal logics etc. In general there are many proof systems that are sound under a given semantics, i.e. there are many sound proof systems for a given logic semantically defined. We present some examples at the end of the chapter. Given a proof system S with logical axioms LA that is sound under a given semantics M. Let $\mathbf{T}_M$ be a set of all tautologies defined by the semantics M, i.e. $\mathbf{T}_M = \{A : \ \models_M A\}$. A natural questions arises: are all tautologies defined by the semantics M, provable in the system S that is sound under the semantics M. The positive answer to this question is called a *completeness* property of the system S. Because we ask the completeness property question for sound systems only we put it in a form of a theorem called a *Completeness Theorem* for a proof system S, under a semantics M. It states that the following holds for any formula A of the language $\mathcal{L}$ of the system S. *A formula A is provable in S if and only if A is a tautology under the semantics M.* We write it symbolically as: $\vdash_S A$ if and only if $\models_M A$. The Completeness Theorem is composed from two parts: the Soundness Theorem and the *completeness part* that proves the completeness property of a sound system.

Proving the Soundness Theorem for S under a semantics M is usually a straightforward and not a very difficult task. We first prove that all logical axioms are **M** tautologies, and then that all inference rules of the system preserve the notion of the **M** truth (**M** model). Proving the *completeness part* of the Completeness Theorem is always a crucial and very difficult task.

We will study two proofs of the Completeness Theorem for classical propositional Hilbert style proof system in chapter **??**, and a constructive proofs for automated theorem proving systems for classical logic the chapter **??**.

Observe that we formulated all these basic theorems linking semantics and syntax (provability) in a general manner. As we first consider propositional languages (chapters **??**, **??**, **??**) and hence we use proof systems for propositional logics as examples. The case of predicate logics will be discussed in chapters ??, ??, ??, ??.

1 Syntax

In this section we formulate a definition of a proof system S by specifying and defining and all its components. We define a notion of a formal proof in a given proof system, and give simple examples of different proof systems. When defining a proof system S we specify, as the first step, its formal language $\mathcal{L}$. When It can be a propositional, or a predicate language. It is a first component of the proof system S. Given a set $\mathcal{F}$ of well formed formulas, of the language $\mathcal{L}$, we often extend this set, and hence the language $\mathcal{L}$ to a set $\mathcal{E}$ of *expressions* build out of the language $\mathcal{L}$, and some additional symbols, if needed. It is a second component of the proof system S. Proof systems act as an inference machine, with provable expressions being its final products. This inference machine is defined by setting, as a starting point a certain non-empty, proper subset LA of $\mathcal{E}$, called a set of *logical axioms* of the system S. The production of provable formulas is to be done by the means of *inference rules*. The inference rules transform an expression, or finite string of expressions, called premisses, into another expression, called conclusion. At this stage the rules don't carry any meaning - they define only how to transform strings of symbols of our language into another string of symbols. This is a reason why investigation of proof systems is called *syntax* or *syntactic investigation* as opposed to *semantcal methods*, which deal with semantics of the language and hence of the proof system. The *syntax- semantics* connection within proof systems is established by Soundness and Completeness theorems and will be discussed in detail in the section 2.

Definition 1 (Proof System)

By a proof system we understand a triple

$$S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R}),$$

where $\mathcal{L} = (\mathcal{A}, \mathcal{F})$ is a formal language, called the language of S with a set $\mathcal{F}$ of formulas; $\mathcal{E}$ is a set of expressions of S; $LA \subseteq \mathcal{E}$ is a non empty set of logical axioms of the system; $\mathcal{R}$ is a finite set of rules of inference.

The components of the proof systems S are defined as follows.

1. The language $\mathcal{L}$ of S

In the propositional case, the formal language $\mathcal{L}$ consists of two components: an alphabet $\mathcal{A}$ and a set $\mathcal{F}$ of formulas. In predicate case the language $\mathcal{L}$ consists of thee components: an alphabet $\mathcal{A}$, a set $\mathbf{T}$ of terms and a set $\mathcal{F}$ of formulas.

The set $\mathbf{T}$ of terms is needed to define properly the set of $\mathcal{F}$ of formulas and we list it as to distinguish it the propositional case. We will denote the language $\mathcal{F}$ of S uniformly as $\mathcal{L} = (\mathcal{A}, \mathcal{F})$ and specify if it is propositional or a predicate language accordingly. We assume that the both sets $\mathcal{A}$ and $\mathcal{F}$ are enumerable, i.e. we will deal here with enumerable languages only.

Semantical Link. Any semantics $\mathbf{M}$ for the language $\mathcal{L}$ is called the semantics for the proof system S.

2. The set $\mathcal{E}$ of expressions of S

Given a set $\mathcal{F}$ of well formed formulas, of the language $\mathcal{L}$, we often extend this set (and hence the language $\mathcal{L}$ to some set $\mathcal{E}$ of *expressions* build out of the language $\mathcal{L}$, and some additional symbols, if needed.

Automated theorem proving systems use as their basic components expressions build out of formulas of the language $\mathcal{L}$. They are for example sequences of formulas in the proof systems **RS** and **RQ** presented in chapter **??** and —in chapter **??**, respectively. The first of such systems Gentzen's systems **LK** for classical logic and **LK** for intuitionistic logic and their followers use expressions called Gentzen sequents. They are presented and discussed in chapter **??**. There also are resolution based proof systems that use different form of expressions to represent for clauses and sets of clauses to mention the few. In many proof system we choose the set of formulas $\mathcal{F}$ as expressions, i.e. we put $\mathcal{E} = \mathcal{F}$.

Semantical Link. We always have to extend a given semantics $\mathbf{M}$ of the language $\mathcal{L}$ of the system S to the set $\mathcal{E}$ of expression.

3. The set LA of logical axioms of S

The logical axioms LA of S form a non-empty subset of the set $\mathcal{E}$ of expressions. In particular, LA is a non-empty subset of formulas, i.e. $LA \subseteq \mathcal{F}$. We assume here that the set LA of logical axioms is finite, i.e. we consider here only *finitely axiomatizable* proof systems.

Semantical Link. Set LA of logical axioms is always a subset of expressions that are *tautologies* under the semantics $\mathbf{M}$ of the language $\mathcal{L}$ of S.

4. The set $\mathcal{R}$ of rules of inference of S

We assume that the proof system S contains a finite number of inference rules. We assume that each rule has a finite number of premisses and one conclusion. We also assume that one can effectively decide, for any inference rule, whether a given string of expressions form its premisses and conclusion or do not, i.e. that all rules $r \in \mathcal{R}$ are primitivvely recursive.

We put it in a formal definition as follows.

Definition 2 (Rule of Inference)

Given a non- empty set $\mathcal{E}$ of expressions of a proof system S. Each rule of inference $r \in \mathcal{R}$ is a relation defined in the set $\mathcal{E}^m$, where $m \geq 1$ with values in $\mathcal{E}$, i.e. $r \subseteq \mathcal{E}^m \times \mathcal{E}$.

Elements $P_1, P_2, \ldots P_m$ of a tuple $(P_1, P_2, \ldots P_m, C) \in r$ are called **premisses** *of the rule r, and C is called its* **conclusion**.

We usually write the inference rules in a following convenient way.

If r is a one premiss rule and $(P_1, C) \in r$, then we write it as

$$(r) \quad \frac{P_1}{C}.$$

If r is a two premisses rule and $(P_1, P_2, C) \in r$, then we write it as

$$(r) \quad \frac{P_1 \quad ; \quad P_2}{C},$$

P_1 is called a left premiss of r and P_2 is called a right premiss.

In general, if r is an m- premisses rule and $(P_1, P_2, ...P_m, C) \in r$, then we will write it as

$$(r) \quad \frac{P_1 \quad ; \quad P_2 \quad ; \quad .... \quad ; \quad P_m}{A}.$$

Semantical Link. We want the rules of inference to preserve truthfulness i.e. to be sound under the semantics $\mathbf{M}$.

Formal Proofs in S

Given a proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$. Final products of a single or multiple use of the inference rules of S, with logical axioms LA taken as a starting point are called provable expressions of the system S. A single use of an inference rule is called a *direct consequence*. A multiple application of rules of inference with axioms taken as a starting point is called a *formal proof*. Formal definitions are as follows.

Definition 3 (DirectConsequence)

A conclusion of a rule of inference is called a direct consequence of its premisses. I.e. for any rule of inference $r \in \mathcal{R}$, if $(P_1, ...P_n, C) \in r$, then C is called a direct consequence of $P_1, ...P_n$ by virtue of r.

Definition 4 (Formal Proof)

Given a proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$. Any sequence $E_1,\ E_2, ,\ E_n$ of expressions from $\mathcal{E}$, such that $n \geq 1$,

$$E_1 \in LA, \qquad E_n = E,$$

and for each $1 < i \leq n$, either $E_i \in LA$ or E_i is a **direct consequence** *of some of the preceding expressions in $E_1,\ E_2, ,\ E_n$ by virtue of one of the rules of inference $r \in \mathcal{R}$ is called a* **formal proof** *of E in S.*

The number $n \geq 1$ is the length of the proof $E_1,\ E_2, ,\ E_n$. We write

$$\vdash_S E$$

to denote that $E \in \mathcal{E}$ **has a formal proof** *in S. When the proof system S is fixed we write $\vdash E$.*

Any expression E such that E has a proof in S, is called a *provable expression* of S. The set of **all provable expressions** of S is denoted by $\mathbf{P}_S$ and is defined as follows.

$$\mathbf{P}_S = \{E \in \mathcal{E} : \ \vdash_S E\}. \tag{1}$$

Consider a simple proof system system S_1 with a language $\mathcal{L} = \mathcal{L}_{\{P,\ \Rightarrow\}}$, where P is one argument connective. We take $\mathcal{E} = \mathcal{F}, LA = \{(A \Rightarrow A)\}$, and the set of rules o inference $\mathcal{R} = \{(r)\ \frac{B}{PB}\}$. We write our proof system as

$$S_1 = (\mathcal{L}_{\{P,\ \Rightarrow\}},\ \mathcal{F},\ \{(A \Rightarrow A)\},\ \ (r)\ \frac{B}{PB}\ ) \tag{2}$$

where A, B are any formulas. Observe that even the system S_1 has only one axiom, it represents an infinite number of formulas. We call such axiom an *axiom schema.*

Consider now a system S_2

$$S_2 = (\mathcal{L}_{\{P,\Rightarrow\}},\ \ \mathcal{F},\ \ \ \{(a \Rightarrow a)\},\ \ \ (r)\ \ \frac{B}{PB}\ ), \tag{3}$$

where $a \in VAR$ is any variable (atomic formula) and $B \in \mathcal{F}$ is any formula. Observe that the system S_2 also has only one axiom similar to the axiom of S_1, both systems have the same rule of inference but they are very different proof systems. For example a formula $((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ is an axiom of the system S_1 for $A = (Pa \Rightarrow (b \Rightarrow c)$ but is not an axiom of the system S_2, as this systems permits axioms of the form: $(a \Rightarrow a)$ for a being a propositional variable.

A formal proof in a system S carries, as the proof system S does, a semantical meaning but it is nevertheless purely *syntactical* in its nature. The rules of inference of a proof system define only how to transform strings of symbols of our language into another string of symbols. The definition of a formal proof says that in order to prove an expression E of a system one has to construct of s sequence of proper transformations as defined by the rules of inference. Here some examples of provable formulas and their formal proofs in both S_1 and S_2 systems. Observe that we do not know the semantics for these systems.

Exercise 1 *Let S_1, S_2 be proof systems (2), (3), respectively. Show that*

$$\vdash_{S_1} ((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$$

$$\vdash_{S_1} P(a \Rightarrow a), \ \ \vdash_{S_2} P(a \Rightarrow a), \ \ \vdash_{S_1} PP(a \Rightarrow a), \ \ \vdash_{S_2} PP(a \Rightarrow a)$$

$$\vdash_{S_1} PPP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)).$$

Solution Formal proof of $((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ in S_1 is one element sequence $A_1 = ((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c))$. It is a proof because the formula A_1 an axiom of S_1. It is not a proof in S_2.

The formulas $P(a \Rightarrow a)$, and $PP(a \Rightarrow a)$ are provable formulas of both proof systems. The formal proofs in both systems of above formulas are identical and are as follows.

Formal proof of $P(a \Rightarrow a)$ in S_1 and S_2 is a sequence $A_1, \ A_2$ for

$$\begin{array}{cc} A_1 = (a \Rightarrow a), & A_2 = P(a \Rightarrow a). \\ \text{axiom} & \text{rule (r) application} \\ & \text{for } B = (a \Rightarrow a) \end{array}$$

Formal proof of $PP(a \Rightarrow a)$ in S_1 and S_2 is a sequence $A_1, \ A_2, \ A_3$ for

$$\begin{array}{ccc} A_1 = (a \Rightarrow a), & A_2 = P(a \Rightarrow a), & A_3 = PP(a \Rightarrow a). \\ \text{axiom} & \text{rule } (r) \text{ application} & \text{rule } (r) \text{ application} \\ & \text{for } B = (a \Rightarrow a) & \text{for } B = P(a \Rightarrow a) \end{array}$$

Formal proof of $PPP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ in S_1 is a sequence $A_1, \ A_2, \ A_3, \ A_4$ for

$$\begin{array}{c} A_1 = ((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c))), \\ \text{axiom} \end{array}$$

$$\begin{array}{c} A_2 = P((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c))), \\ \text{rule } (r) \text{ application} \end{array}$$

$$\begin{array}{c} A_3 = PP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c))), \\ \text{rule } (r) \text{ application} \end{array}$$

$$\begin{array}{c} A_4 = PPP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c))). \\ \text{rule } (r) \text{ application} \end{array}$$

It is not a proof in S_2. Moreover

$$\nvdash_{S_2} PPP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)).$$

Observe that even if the set of axioms and the inference rules of the proof system are primitively recursive it doesn't mean that the notion of "provable expression" is also primitively recursive, i.e. that there always will be an effective, mechanical method (effective procedure) for determining, given any expression A of the system, whether there is a proof of A. We define the following notions

Definition 5 (Decidable system)

A proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$ for which there is an effective decision procedure for determining, for any expression E of the system, whether there is, or there is no proof of E in S is called a **decidable proof system***, otherwise it is called* **undecidable***.*

Observe that the above notion of decidability of the system S does not require us to find a proof, it requires only a mechanical procedure of deciding whether *there is*, or there is no such a proof. We hence introduce a following notion.

Definition 6 (Syntactic Decidability)

A proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$ for which there is an effective mechanical, procedure that finds (generates) a formal proof of any E in S, if it exists, is called **syntactically semi- decidable***. If additionally there is an effective method of deciding that if a proof of E in S not found, it does not exist, the system S is called* **syntactically decidable***. Otherwise S is* **syntactically undecidable***.*

The existence of prove systems for classical logic and mathematics that are syntactically decidable or syntactically semi-decidable was stated (in a different form) by German mathematician David Hilbert in early 1900 as a part of what is called Hilbert's program. The main goal of Hilbert's program was to provide secure foundations for all mathematics. In particular it addressed the problem of decidability; it said that here should be an algorithm for deciding the truth or falsity of any mathematical statement. Moreover, it should use only "finitistic" reasoning methods. Kurt Gdel showed in 1931 that most of the goals of Hilbert's program were impossible to achieve, at least if interpreted in the most obvious way. Nevertheless, Gerhard Gentzen in his work published in 1934/1935 gave a positive answer to existence of syntactical decidability. He invented proof systems for classical and intiutionistic logics, now called Gentzen style formalizations. They formed a basis for development of Automated Theorem Proving area of mathematics and computer science. We will study the Gentzen style formalizations in chapter **??**.

Definition 7 (Automated Systems)

A proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$ that is proven to be syntactically decidable or semi-decidable is called an automated proof systems.

Automated proof systems are also called *automated theorem proving systems, Gentzen style formalizations, syntactically decidable systems* and and we use all of these terms interchangeably.

Example 1 *Any complete Hilbert style proof system for classical propositional logic is an example of a decidable, but not syntactically decidable proof system. We conclude its decidability from the Completeness Theorem (to be proved in next chapter) and the decidability of the notion of classical tautology (proved in chapter 3).*

Example 2 *The Gentzen style proof systems for classical and intuiionistic propositional logics presented in chapter* **??**, *are examples of proof systems that are of both decidable and syntactically decidable.*

W are going to prove now, as a simple example t the following

Fact 1

The systems proof systems S_1 and S_2 defined by (2) and (3), respectively are syntactically decidable.

Proof Let's now to think how we can search for a proof in S_2 of a formula

$$PP((Pa \Rightarrow (b \Rightarrow c)).$$

If $PP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ had the proof, the only last step in this proof would have been the application of the rule (r) to the formula $PP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$. This formula, in turn, if it had the proof, the only last step in its proof would have been the application of the rule r to the formula $P((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$. And again, this one could be obtained only from the formula $((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ by the virtue of the rule r. Here the search process stops; the rule r puts P in front of the formulas, hence couldn't be applied here. The formula $((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ isn't an axiom of S_2, what means that the only possible way of finding the proof has failed, i.e. we have proved that $\nvdash_{S_1} PPP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$.

The above example of proof search in S_2 defines the following an effective, automatic **Procedure** S_1, S_2 of searching for a proof of our formula in both our proof systems. If the search ends with an axiom, we have a proof, if it doesn't end with an axiom it means that the proof does not exists. We have described it, as an example, for one particular formula. It can be easily extended to any formula A of $\mathcal{L}_{\{P, \Rightarrow\}}$ as follows.

Procedure S_1, S_2

Step : Check the main connective of A.

If main connective is P, it means that A was obtained by the rule r.

Erase the main connective P.

Repeat until no P left.

If the main connective is $\Rightarrow$,check if a formula A is an axiom.

If it is an axiom , STOP and YES, we have a proof.

If it is not an axiom , STOP and NO, proof does not exist.

It is an effective, automatic procedure of searching for a proof of our formula in both our proof systems. This **ends the proof** .

Observe also, that the systems S_1 and S_2 are such that we can easily describe a general form of their provable formulas defined by (1) as $\mathbf{P}_S = \{E \in \mathcal{E} : \ \vdash_S E\}$. Namely we have the following.

$$\mathbf{P}_{S_1} = \{P^n(A \Rightarrow A) : \quad n \in N, \quad A \in \mathcal{F}\},$$

$$\mathbf{P}_{S_2} = \{P^n(a \Rightarrow a) : \quad n \in N, \quad a \in VAR\},$$

where P^n denotes n-iteration of P for $n \geq 1$ and P^0 denotes absence of P.

Obviously we have that $\mathbf{P}_{S_1} \neq \mathbf{P}_{S_2}$, and $\mathbf{P}_{S_2} \subseteq \mathbf{P}_{S_1}$.

The proof systems S_1 and S_2 are very simple, indeed. Here is an example of another two, similar but slightly more complex proof systems.

Consider two proof systems S_3 and S_4 of the language $\mathcal{L}_{\{\cup,\neg\}}$ with the set of expressions $\mathcal{E} = \mathcal{F}$ and is defined as follows.

$$S_3 = (\mathcal{L}_{\{\cup,\neg\}},\ \mathcal{F}, \quad \{(A \cup \neg A)\}, \ (r)\ \frac{(A \cup \neg A)}{(B \cup (A \cup \neg A))},\ \text{for any } A, B \in \mathcal{F}\ ). \quad (4)$$

$$S_4 = (\mathcal{L}_{\{\cup,\neg\}},\ \ \mathcal{F},\ \{(A \cup \neg A)\},\ \ (r)\ \frac{B}{(B \cup (A \cup \neg A))},\ \text{for any } A, B \in \mathcal{F}\ ), \quad (5)$$

Exercise 2 *Given proof systems S_3 and S_4 defined by (4), (5), respectively.*

1. Describe the sets $\mathbf{P}_{S_3}, \mathbf{P}_{S_4}$ of provable formulas of S_3 and S_4.

2. Decide whether is it true/ false *that $\mathbf{P}_{S_3} = \mathbf{P}_{S_4}$. If yes, prove it, if not, give an example of a formula A such that $A \in \mathbf{P}_{S_4}$ and $A \notin \mathbf{P}_{S_3}$, or vice versa.*

Solution 1.
Let's first describe the set of provable formulas of both systems. Consider proof system S_3. Obviously, for any formula $A \in \mathcal{F}$, $(A \cup \neg A)$, as it is the axiom. It constitutes a proof of length 1 $A_1 = (A \cup \neg A)$and we have that

$$\vdash_{S_3} (A \cup \neg A).$$

One application of the inference rule (r) to axiom$(A \cup \neg A)$ gives us a proof $A_1 = (A \cup \neg A),\ A_2 = ((A \cup \neg A) \cup (A \cup \neg A))$, and hence

$$\vdash_{S_3} ((A \cup \neg A) \cup (A \cup \neg A)).$$

The application of the rule (r) to the already proven above formula A_2 give us the proof $A_1 = (A \cup \neg A),\ A_2 = ((A \cup \neg A) \cup (A \cup \neg A)), A_3 = (((A \cup \neg A) \cup (A \cup \neg A)) \cup (A \cup \neg A))$, and

$$\vdash_{S_3} (((A \cup \neg A) \cup (A \cup \neg A)) \cup (A \cup \neg A)).$$

It is easy to see that all provable formulas of S_3 will be of the form of the proper disjunction of the axiom of S_3, what we denote as follows:

$$\mathbf{P}_{S_3} = \{\bigcup_{n \in N} (A \cup \neg A)^n : A \in \mathcal{F}\}, \tag{6}$$

where $(A \cup \neg A)^n$ denotes a disjunction of n formulas of the form $(A \cup \neg A)$ defined recursively as follows. $(A \cup \neg A)^0 = (A \cup \neg A)$, $(A \cup \neg A)^{n+1} = ((A \cup \neg A)^n \cup (A \cup \neg A))$.

Consider now system S_4. Obviously, as in the case of S_4, $\vdash_{S_4} (A \cup \neg A)$. One application of the inference rule to the axiom gives us the proof $A_1 = (A \cup \neg A),\ A_2 = (B \cup (A \cup \neg A))$ and we have that

$$\vdash_{S_4} (B \cup (A \cup \neg A)), \tag{7}$$

where B can be any formula from $\mathcal{F}$.

The rule (r) can't be, by its definition, applied to already proved $B \cup (A \cup \neg A))$. We can continue with the proof A_1, A_2 by constructing for example a proof A_1, A_2, A_3, A_4 by inserting axiom $(C \cup \neg C)$ (or axiom $(A \cup \neg A)$, if we wish as A_3 step of the proof. We have to remember that the definition 4 of the formal proof allows us to insert an axiom in any place within the proof. $A_1 = (A \cup \neg A),\ A_2 = (B \cup (A \cup \neg A)),\ A_3 = (C \cup \neg C),\ A_4 = (A \cup (C \cup \neg C))$ and hence

$$\vdash_{S_4} (A \cup (C \cup \neg C)),\ \vdash_{S_4} (B \cup (A \cup \neg A)), \vdash_{S_4} (C \cup (B \cup \neg B)), \text{......etc...}$$

Multiple application of the rule (r) in S_4 means its application to multiple forms of the axiom. Finally it is clear that we can only construct formal proofs of all possible formulas of the form $(B \cup (A \cup \neg A))$, and of course of a form of any axiom (proofs of the length 1) $(A \cup \neg A)$ for A, B being all possible formulas. Remark that by saying $A, B \in \mathcal{F}$ we do not say that $A \neq B$, that we do not exclude that case $A = B$. In particular case we have that

$$\vdash_{S_4} (A \cup (A \cup \neg A)),\ \vdash_{S_4} (B \cup (B \cup \neg B)), \vdash_{S_4} (C \cup (C \cup \neg C)), ...\text{etc...}$$

Hence

$$\mathbf{P}_{S_4} = \{(B \cup (A \cup \neg A)) : \quad A, B \in \mathcal{F}\} \cup \{(A \cup \neg A) : \quad A \in \mathcal{F}\}. \tag{8}$$

Solution 2.
We prove now that $\mathbf{P}_{S_3} \subseteq \mathbf{P}_{S_4}$. Let $D \in \mathbf{P}_{S_3}$. By (6) $D = \bigcup_{n \in N} (A \cup \neg A)^n$. Observe that by definition $D = \bigcup_{n \in N} (A \cup \neg A)^n = (\bigcup_{n \in N} (A \cup \neg A)^{n-1} \cup (A \cup \neg A))$ and $\bigcup_{n \in N} (A \cup \neg A)^{n-1}$ is a formula of $\mathcal{L}_{\{\cup, \neg\}}$. We can denote it by B. We have proved in (7) that for any $B \in \mathcal{F}$, $\vdash_{S_4} (B \cup (A \cup \neg A))$. But by definition $D = (B \cup (A \cup \neg A))$, hence we proved that $D \in \mathbf{P}_{S_4}$.This ends the proof.

Consider a formula $((a \cup \neg b) \cup (a \cup \neg a))$ of $\mathcal{L}_{\{\cup, \neg\}}$. It has a following formal proof A_1, A_2 in S_4.

$$\begin{array}{cc} A_1 = (a \cup \neg a), & A_2 = ((a \cup \neg b) \cup (a \cup \neg a)). \\ \text{axiom for A=a} & \text{rule (r) application} \\ & \text{for } B = (a \cup \neg b) \end{array}$$

This proves that $((a \cup \neg b) \cup (a \cup \neg a)) \in \mathbf{P}_{S_4}$. Obviously $\not\vdash_{S_3} ((a \cup \neg b) \cup (a \cup \neg a))$ and $((a \cup \neg b) \cup (a \cup \neg a)) \notin \mathbf{P}_{S_3}$. We have proved that the proof systems S_3 and S_4 defined by (4), (5) are such that $\mathbf{P}_{S_3} \subseteq \mathbf{P}_{S_4}$ and $\mathbf{P}_{S_3} \neq \mathbf{P}_{S_4}$.

Consider now a following proof system S_5.

$$S_5 = (\ \mathcal{L}_{\{\Rightarrow, \cup, \neg\}},\ \mathcal{F},\ \{(A \Rightarrow (A \cup B))\},\ \ \{(r1),\ (r2)\}\ ) \tag{9}$$

where the rules of inference are defined as follows.

$$(r1)\ \frac{A\ ; B}{(A \cup \neg B)}, \qquad (r2)\ \frac{A\ ; (A \cup B)}{B}.$$

Exercise 3

Given proof systems S_5 defined by (9).

1. Find a formal proof of a formula $\neg(A \Rightarrow (A \cup B))$ in S_5, i.e. show that $\vdash_{S_5} \neg(A \Rightarrow (A \cup B))$.

2. Find a formal proof of a formula $\neg((a \cup \neg b) \Rightarrow ((a \cup \neg b) \cup (a \cup \neg a)))$.

Solution

1. We construct a formal proof $B_1,\ B_2,\ B_3,\ B_4$ of the formula $\neg(A \Rightarrow (A\cup B))$ as follows. We write comments next to each step of the proof.

$B_1 = (A \Rightarrow (A \cup B))$ axiom, $\quad B_2 = (A \Rightarrow (A \cup B))$ axiom,

$B_3 = ((A \Rightarrow (A \cup B)) \cup \neg(A \Rightarrow (A \cup B)))$ rule (r1) application to B_1 and B_2,

$B_4 = \neg(A \Rightarrow (A \cup B))$ rule (r2) application to B_3

for $A = (A \Rightarrow (A \cup B))$ and $B = \neg(A \Rightarrow (A \cup B))$.

2. We construct a formal proof $B_1,\ B_2,\ B_3,\ B_4$ of the formula $\neg((a \cup \neg b) \Rightarrow ((a \cup \neg b) \cup (a \cup \neg a)))$ as follows.

$B_1 = ((a\cup\neg b) \Rightarrow ((a\cup\neg b)\cup(a\cup\neg a)))$ axiom for $A = (a\cup\neg b)$ and $B = (a\cup\neg a)$,

$B_2 = ((a\cup\neg b) \Rightarrow ((a\cup\neg b)\cup(a\cup\neg a)))$ axiom for $A = (a\cup\neg b)$ and $B = (a\cup\neg a)$,

$B_3 = (((a \cup \neg b) \Rightarrow ((a \cup \neg b) \cup (a \cup \neg a))) \cup \neg((a \cup \neg b) \Rightarrow ((a \cup \neg b) \cup (a \cup \neg a))))$ rule (r1) application to B_1 and B_2,

$B_4 = \neg((a \cup \neg b) \Rightarrow ((a \cup \neg b) \cup (a \cup \neg a)))$ rule (r2) application to B_3

for $A = ((a \cup \neg b) \Rightarrow ((a \cup \neg b) \cup (a \cup \neg a)))$ and $B = \neg((a \cup \neg b) \Rightarrow ((a \cup \neg b) \cup (a \cup \neg a)))$.

Observation 1 *Observe that the formula $\neg((a\cup\neg b) \Rightarrow ((a\cup\neg b)\cup(a\cup\neg a)))$ is a particular case of the formula $\neg(A \Rightarrow (A\cup B))$ for $A = (a\cup\neg b)$ and $B = (a\cup\neg a)$ and its proof is just a particular case of the proof constructed in case 1.*

We wrote down independently a complicated proof of the particular case to make reader aware of a need of generalizing particular formulas, if it possible, and writing simpler proofs for the general case instead of the perticular.

Consequence Operation

Given a proof system $S = (\mathcal{L},\ \mathcal{E},\ LA, \mathcal{R})$. While proving expressions we often use some extra information available, besides the axioms of the proof system. This extra information is called *hypotheses* in the proof.

Let $\Gamma \subseteq \mathcal{E}$ be a set expressions called hypotheses. A proof from a *set* Γ *of hypothesis* of an expression $E \in \mathcal{E}$ in $S = (\mathcal{L}, LA, \mathcal{R})$ is a formal proof in S, where the expressions from Γ are treated as additional hypothesis added to the set LA of the logical axioms of the system S. We define it formally as follows.

Definition 8 (Proof from Hypotheses)

Given a proof system $S = (\mathcal{L},\ \mathcal{E},\ LA, \mathcal{R})$ and let Γ be any set of expressions of S, i.e. let $\Gamma \subseteq \mathcal{E}$.

A proof of an expression $E \in$ from the set Γ of expressions is a sequence

$$E_1, E_2,\ \dots\ E_n$$

of expressions, such that

$$E_1 \in LA \cup \Gamma, \qquad E_n = E$$

and for each i, $1 < i \leq n$, either $E_i \in LA \cup \Gamma$ or E_i is a direct consequence of some of the preceding expressions in the sequence $E_1, E_2,\ \dots\ E_n$ by virtue of one of the rules of inference from $\mathcal{R}$.

We write

$$\Gamma \vdash_S E$$

to denote that the expression E has a proof from Γ in S and $\Gamma\ \vdash\ E$, when the system S is fixed.

When the set of hypothesis Γ is a *finite set* and $\Gamma = \{B_1, B_2, ..., B_n\}$, then we write

$$B_1, B_2, ..., B_n \vdash_S E$$

instead of $\{B_1, B_2, ..., B_n\} \vdash_S E$. The case when Γ is an empty set i.e. when $\Gamma = \emptyset$ is a special one. By the definition of a proof of E from Γ, $\emptyset \vdash E$ means that in the proof of E only logical axioms LA of S were used. We hence write it as we did before

$$\vdash_S E$$

to denote that E has a proof from the empty set Γ. The set of all expressions provable from Γ (and logical axioms LA in S is denoted by $\mathbf{P}_S(\Gamma)$, i.e.

$$\mathbf{P}_S(\Gamma) = \{E \in \mathcal{E} :\quad \Gamma \vdash_S E\}. \tag{10}$$

When discussing properties of provability in proof systems we often use a notion of a *consequence operation*. In order to follow this tradition we call provable expressions from Γ in S *consequences* of Γ. The set of all expressions provable is then called the *set of all consequences* from Γ. We observe that when talking about consequences of Γ in S, we define in fact a function which to every set $\Gamma \subseteq \mathcal{E}$ assigns a set of all its consequences. We denote this function by $\mathbf{Cn}_S$ and adopt the following definition.

Definition 9 (Consequence)

Given a proof system $S = (\mathcal{L},\ \mathcal{E},\ LA, \mathcal{R})$. Any function $\mathbf{Cn}_S\ : 2^{\mathcal{E}} \longrightarrow 2^{\mathcal{E}}$ such that for every $\Gamma \in 2^{\mathcal{E}}$,

$$\mathbf{Cn}_S(\Gamma) = \{E \in \mathcal{E} :\quad \Gamma \vdash_S E\} \tag{11}$$

is called a **consequence determined by** *S.*

Directly from definition 14 and (10) we have the following.

Fact 2 *For any proof system $S = (\mathcal{L},\ \mathcal{E},\ LA, \mathcal{R})$.*

$$\mathbf{P}_S(\Gamma) = \mathbf{Cn}_S(\Gamma). \tag{12}$$

It proves that the notions of provability from a set Γ in S and consequence determined by S coincide . It means that we can, and we often use in the literature the both terms interchangeably.

The definition 14 does do more then just re-naming "provability" by "consequence". We are going to prove now that the consequence $\mathbf{Cn}_S$ determined by S a special (an important) case of a notion a classic consequence operation as defined by Alfred Tarski in 1930 as a general model of deductive reasoning. Tarski definition is a formalization of the intuitive concept of the deduction as a consequence, and therefore it has all the properties which our intuition attribute to this notion. Here is the definition.

Definition 10 (Tarski)

By a **consequence operation** *in a formal language $\mathcal{L} = (\mathcal{A}, \mathcal{F})$ we understand any mapping $\mathbf{C} : 2^{\mathcal{F}} \longrightarrow 2^{\mathcal{F}}$ satisfying the following conditions (t1) - (t3) expressing properties of reflexivity, monotonicity, and transitivity of the consequence.*

For any sets $F,\ F_0,\ F_1,\ F_2,\ F_3 \in 2^{\mathcal{F}}$,

(t1) $F \subseteq \mathbf{C}(F)$ **reflexivity**,

(t2) if $F_1 \subseteq F_2$, *then* $\mathbf{C}(F_1) \subseteq \mathbf{C}(F_2)$, **monotonicity**.

(t3) if $F_1 \subseteq \mathbf{C}(F_2)$ *and* $F_2 \subseteq \mathbf{C}(F_3)$, *then* $F_1 \subseteq \mathbf{C}(F_3)$, **transitivity**.

We say that the consequence operation $\mathbf{C}$ *has a* **finite character** *if additionally it satisfies the following condition* **t4**.

(t4) if a formula $B \in \mathbf{C}(F)$, *then there exists a finite set* $F_0 \subseteq F$, *such that* $B \in \mathbf{C}(F_0)$. **finiteness**.

The monotonicity condition (t2) and transitivity condition (t3) are often replaced by the following conditions **(t2')**, **(t3')**, respectively.

For any formula $B \in \mathcal{F}$, any any sets $F,\ F', \in 2^{\mathcal{F}}$,

$$(t2') \quad \text{if } B \in \mathbf{C}(F), \quad then\ B \in \mathbf{C}(F \cup F'), \tag{13}$$

$$(t3') \quad \mathbf{C}(F) = \mathbf{C}(\mathbf{C}(F)). \tag{14}$$

We express the correctness of the replacement conditions (13) and (14) in a form of a following theorem.

Theorem 1

The Tarski definition 10 is equivalent with definitions where one, or both conditions (t2), (t3) are replaced respectively by conditions (t2'), (t3')given by equations (13) and (14).

Proof We prove the equivalency of conditions (t1) - (t3) and (t1) - (t3'). We leave the proof of the other equivalency to the reader.

Assume (t3). By substituting

$$F_1 = \mathbf{C}(\mathbf{C}(F)),\ F_2 = \mathbf{C}(F),\ F_3 = F$$

in (t3) we obtain

$$\mathbf{C}(\mathbf{C}(F)) \subseteq \mathbf{C}(F).$$

On the other hand, it follows from (t1) and (t2)

$$\mathbf{C}(F) \subseteq \mathbf{C}(\mathbf{C}(F)),$$

which with the previous inclusion gives (t3'). Conversely, suppose that (t3') is satisfied. If $F_2 \subseteq \mathbf{C}(F_3)$, then by (t2) we obtain $\mathbf{C}(F_2) \subseteq \mathbf{C}(\mathbf{C}(F_3))$. By (t3') $\mathbf{C}(\mathbf{C}(F_3)) = (\mathbf{C}(F_3)$, hence $\mathbf{C}(F_2) \subseteq (\mathbf{C}(F_3)$ and we proved (t3).

The consequence operation provides a model describing what we intuitively call a deduction. It formalizes the basic, intuitively obvious and accepted properties of reasoning by which we obtain (deduce) new facts from already known, or assumed. We hence use it to define, after Tarski, a following notion of a deductive system.

Definition 11 (Deductive System)

Given a formal language $\mathcal{L} = (\mathcal{A}, \mathcal{F})$ and a Tarski consequence $\mathbf{C}$ (definition 10). A system

$$D = (\mathcal{L}, \mathbf{C})$$

is called a Tarski deductive system for the language $\mathcal{L}$.

Tarski's deductive system as a model of reasoning does not provide a method of actually defining a consequence operation in the language $\mathcal{L}$; it assumes that it is given. We are going to prove now that our definition 14 of consequence operation $\mathbf{Cn}_S$ determined by a proof system S is a Tarski consequence operation $\mathbf{C}$ in the definition 10 sense. It justifies, together with Fact 2 the common use the consequence notion when talking about provability. It means that each proof system S provides a different example of a consequence operation. They are all purely syntactic in nature and all defined by the notion of provability. Hence each proof system can be treated and a syntactic Tarski deductive system from definition 11.

Theorem 2

Given a proof system $S = (\mathcal{L},\ \mathcal{E},\ LA, \mathcal{R})$. The consequence operation $\mathbf{Cn}_S$ *is a Tarski consequence* $\mathbf{C}$ *in the language* $\mathcal{L}$ *of the system* S *and the system*

$$D_S = (\mathcal{L},\ \mathbf{Cn}_S)$$

is Tarski deductive system. We call it a **syntactic deductive system** *determined by S. Moreover, the consequence operation* $\mathbf{Cn}_S$ *that has a* **finite character**.

Proof

By definition 14, the consequence operation $\mathbf{Cn}_S\ : 2^{\mathcal{E}} \longrightarrow 2^{\mathcal{E}}$ is given by a formula $\mathbf{Cn}_S(\Gamma) = \{E \in \mathcal{E} :\ \Gamma \vdash_S E\}$. We have to show that for any $\Gamma,\ \Gamma_0,\ \Gamma_1,\ \Gamma_2,\ \Gamma_3 \in 2^{\mathcal{F}}$ conditions (t1) - (t4) of the definition 14 hold. The reflexivity condition (t1) becomes $\Gamma \subseteq \mathbf{Cn}_S(\Gamma)$. Let $E \in \Gamma$. The one element sequence E is a proof of E from Γ, hence we proved that $E \in\ \mathbf{C}(\Gamma)$ and (t1) holds. To prove the transitivity condition (t2) assume now that $\Gamma_1 \subseteq \Gamma_2$. Let $E \in \mathbf{Cn}_S(\Gamma_1)$. It means that $\Gamma_1 \vdash_S E$, i.e E has a formal proof from Γ_1, but $\Gamma_1 \subseteq \Gamma_2$, hence this proof also is a proof from Γ_2, and $E \in \mathbf{Cn}_S(\Gamma_2)$. This proves that $\mathbf{Cn}_S(\Gamma_1) \subseteq \mathbf{Cn}_S(\Gamma_2)$ and the condition (t2) holds. Let now $E \in \Gamma_1$ and $\Gamma_1 \subseteq \mathbf{Cn}_S(\Gamma_2)$, so $E \in \mathbf{Cn}_S(\Gamma_)2$. Let $E_1, \ldots, E_n$ be a formal proof of E from Γ_2. But $\Gamma_2 \subseteq \mathbf{Cn}_S(\Gamma_3)$. It means that any expression from Γ_2 has a formal proof from Γ_3. In particular, all expression in the proof $E_1, \ldots, E_n$ that belong to Γ_2 have their formal proofs from Γ_3. Replacing all these expressions by their proofs from Γ_3 we obtain a proof of E from Γ_3. This proves that $\Gamma_1 \subseteq \mathbf{Cn}_S\Gamma_3$ and the transitivity condition (t3) holds. Let now $E \in \mathbf{Cn}_S\Gamma$. This means that E has a proof $E_1, \ldots, E_n$ from Γ. The set $\Gamma_0 = \{E_1, \ldots, E_n\}$ is obviously a finite subset of Γ and $E \in \mathbf{Cn}_S\Gamma_0$ and (t4) holds.

Non - Monotonic Logics

The Tarski definition 10 of a consequence models reasoning which is called after its condition (t2) or (t2') a monotonic reasoning. The monotonicity of reasoning was, since antiquity is the the basic assumption while developing models for classical and well established non-classical logics. Recently many of new non-classical logics were developed and are being developed by computer scientists. For example new modal logics of agents and temporal logics. Temporal logics are essential in developing theory of programs. Nevertheless they all are built following the Tarski definition of consequence and were and are called *monotonic logics*. A new type of important

Non-Monotonic logics have been proposed at the beginning of the 80s. Historically the most important proposals are Non-monotonic logic, by McDermott and Doyle, Default Logic, by Reiter, Circumscription, by McCarthy, and Autoepistemic logic, by Moore.

The term non-monotonic logic covers a family of formal frameworks devised to capture and represent defeasible inference. It is an inference in which it is possible to draw conclusions tentatively, reserving the right to retract them in the light of further information. We included most standard examples in Chapter 1, Introduction.

Non-monotonic logics describe commonsense reasoning which is neither a restriction nor an extension of classical logic. Consequences of premises are drawn as much due to the absence as to the presence of knowledge. When more knowledge is acquired, conclusions previously drawn may have to be withdrawn because the rules of inference that led to them no longer are active. Intelligent decision makers use this form of commonsense reasoning to infer actions to be performed from premises which cannot be made by classical logic inference, because they simply have to make decisions whether or not there is enough information for a classical logical deduction. Non-monotonic reasoning deals with the problem of deriving plausible conclusions, but not infallible, from a knowledge base (a set of formulas). Since the conclusions are not certain, it must be possible to retract some of them if new information shows that they are wrong. Example: let the KB contain: Typically birds fly. Penguins do not fly. Tweety is a bird. It is plausible to conclude that Tweety flies. However if the following information is added to KB Tweety is a penguin the previous conclusion must be retracted and, instead, the new conclusion that Tweety does not fly will hold.

The statement "typically A" can be read as: "in the absence of information to the contrary, assume A". The problem is to define the precise meaning of "in the absence of information to the contrary". The meaning could be: "there is nothing in KB that is inconsistent with assumption A". Other interpretations are possible Different interpretations give rise to different non-monotonic logics.

Formal Theories

Formal theories play crucial role in mathematics and were historically defined for classical first order logic and consequently for other first and higher order logics. They are routinely called *first order theories.* We will discuss them in more detail in chapter **??** dealing formally with classical predicate logic. First order theories are hence based on proof systems S with a predicate (first order) language $\mathcal{L}$. We will call them for short *first order proof systems.*

We can and we sometimes consider formal theories based on propositional logics, i.e. based on proof systems with language $\mathcal{L}$ being propositional. We will call them *propositional theories.*

Given a proof system $S = (\mathcal{L},\ \mathcal{E},\ LA,\ \mathcal{R})$. We build (define) a **formal theory** based on S as follows.

1. We select a certain *finite subset* SA of expressions of S, disjoint with the logical axioms LA of S, i.e. such that $LA \cap SA = \emptyset$.. The set SA is called a set of *specific axioms* of the formal theory based on S.

2. We use set SA of specific axioms to define a language

$$[\mathcal{L}_{SA}, \tag{15}$$

called a *language of the formal theory*. Here we have two cases.

c1. S is a *first order proof system*, i.e. $\mathcal{L}$ of S is a predicate language. We define the language $\mathcal{L}_{SA}$ by restricting the sets of constant, functional, predicate symbols of $\mathcal{L}$ to constant, functional, predicate symbols appearing in the set SA of specific axioms. Both languages $\mathcal{L}_{SA}$ and $\mathcal{L}$ share the same set of *propositional connectives.*

c2. S is a *propositional proof system*, i.e. $\mathcal{L}$ of S is a propositional language. $\mathcal{L}_{SA}$ is defined by restricting $\mathcal{L}$ to connectives appearing in the set SA.

Definition 12 (Formal Theory)

The system

$$T = (\mathcal{L},\ \mathcal{E},\ LA,\ SA,\ \mathcal{R}) \tag{16}$$

is called a **formal theory** *based on a proof system S.*

The set SA of the set of **specific axioms** *of T. The language $\mathcal{L}_{SA}$ defined by (15) is called the* **language of the theory** *T.*

The set $\mathcal{E}_{SA}$ of all expressions of the language $\mathcal{L}_{SA}$ provable from the set specific axioms SA (and logical axioms LA) i.e. the set

$$\mathbf{T}(SA) = \{E \in \mathcal{E}_{SA} : \quad SA \ \vdash_S E\ \} \tag{17}$$

is called the set of all **theorems** of the theory T.

If the set SA of specific axioms of T is empty, then the theory T is, by definition, identified with the system S, i.e. $T = S = (\mathcal{L},\ \mathcal{E},\ LA,\ \mathcal{R})$.

Definition 13 (Consistent Theory)

A theory $T = (\mathcal{L},\ \mathcal{E},\ LA,\ SA,\ \mathcal{R})$ is **consistent** *if there exists an expression $E \in \mathcal{E}_{SA}$ such that $E \notin \mathbf{T}(SA)$, i.e. such that*

$$SA \ \not\vdash_S E;$$

otherwise the theory T is **inconsistent**.

Observe that the definition 13 has purely syntactic meaning. It also reflexes our intuition what proper provability should mean. it says that a formal theory T based on a proof system S is consistent only when it does not prove all expressions (formulas in particular cases) of $\mathcal{L}_{SA}$. The theory T such that it proves everything stated in $\mathcal{L}_{SA}$ obviously should be, and its defined as inconsistent. In particular, we have the following *syntactic definition* of consistency-inconsistency for any proof system S.

Definition 14 (Syntactic Consistency)

A proof system $S = (\mathcal{L},\ \mathcal{E},\ LA,\ \mathcal{R})$ is **consistent** *if there exists $E \in \mathcal{E}$ such that $E \notin \mathbf{P}_S$, i.e. such that $\nvdash_S E$; otherwise S is* **inconsistent***.*

2 Semantics

We define formally a semantics for a given proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$ by specifying the semantic links of all its components as follows.

Semantic Link1: Language $\mathcal{L}$

The language $\mathcal{L}$ of S can be propositional or predicate. Let denote my $\mathbf{M}$ a semantic for L. We call it, for short, a semantics for the proof system S. The semantics $\mathbf{M}$ can be a propositional, a predicate, extensional, or not extensional. We use $\mathbf{M}$ as a general symbol for a semantics.

Semantic Link 2: Set $\mathcal{E}$ of Expressons

We always have to extend a given semantics $\mathbf{M}$ of the language $\mathcal{L}$ to the set of expressions $\mathcal{E}$ of the system S. We often do it by establishing a semantic equivalency under semantics $\mathbf{M}$ of $\mathcal{E}$ and the set of all formulas $\mathcal{F}$ of $\mathcal{L}$. It means we prove that for a given semantics M under which we build our proof system S, and for any expression $E \in \mathcal{E}$ there is a formula $A \in \mathcal{F}$, such that $E {\equiv}_M A$. For example, in the automated theorem proving system **RS** presented in chapter **??** the expressions are *finite sequences of formulas* of $\mathcal{L} = \mathcal{L}_{\neg,\cap,\cup,\Rightarrow}$. We extend our classical semantics for $\mathcal{L}$ to the set $\mathcal{F}^*$ of all finite sequences of formulas as follows: for any $v : VAR \longrightarrow \{F, T\}$ and any $\Delta \in \mathcal{F}^*$, $\Delta = A_1, A_2, ..A_n$, $v^*(\Delta) = v^*(A_1, A_2, ..A_n) = v^*(A_1) \cup v^*(A_2) \cup .... \cup v^*(A_n)$, i.e. $\Delta \equiv (A_1 \cup A_2 \cup ... \cup A_n)$. Sometimes, like in case of Resolution based proof systems we have also to prove a semantic equivalency of a given formula A of $\mathcal{L}$ with some set $\mathcal{E}_A$ of expressions (sets of clauses) representing the formula A.

Semantic Link 3: Logical Axioms LA

Given a semantics $\mathbf{M}$ for $\mathcal{L}$ and its extension to the set $\mathcal{E}$ of all expressions. We extend the notion of tautology to the set $\mathcal{L}$ of expressions and write $\models_{\mathbf{M}} E$ to denote that the expression $E \in \mathcal{E}$ is a tautology under semantics $\mathbf{M}$. We denote

$$\mathbf{T}_{\mathbf{M}} = \{E \in \mathcal{E} : \ \models_{\mathbf{M}} E\}$$

While designing a proof system S we want the logical axioms LA to be a subset of expressions that are tautologies of under the semantics $\mathbf{M}$, i.e.

$$LA \subseteq \mathbf{T_M}.$$

We can, and we often do, invent proof systems with languages without yet established semantics. In this case the logical axioms LA serve as description of properties of tautologies under a future semantics yet to be built. We want to choose as logical axioms of a proof system S are not only tautologies under an already known semantics $\mathbf{M}$, but they can also guide us how to define a semantics when it is yet unknown.

Semantic Link 4: Rules of Inference $\mathcal{R}$

We want the rules of inference $r \in \mathcal{R}$ to preserve truthfulness. Rules that preserve the truthfulness are called sound under a given semantics $\mathbf{M}$. We put it in a general formal definition as follows.

Definition 15 (Sound Rule under M)

Given an inference rule $r \in \mathcal{R}$ of the form

$$(r) \qquad \frac{P_1 \;\; ; \;\; P_2 \;\; ; \;\; .... \;\; ; \;\; P_m}{C}.$$

We say that the rule (r) is **sound** *under a semantics* $\mathbf{M}$ *if the following condition holds for all* $\mathbf{M}$ *models* $\mathcal{M}$.

$$\textit{If} \;\; \mathcal{M} \models_{\mathbf{M}} \{P_1, P_2, .P_m\} \;\; \textit{then} \;\; \mathcal{M} \models_{\mathbf{M}} C. \tag{18}$$

In case of a propositional language $\mathcal{L}_{CON}$ and an extensional semantics $\mathbf{M}$ the $\mathbf{M}$ models $\mathcal{M}$ are defined in terms of the truth assignment $v : VAR \longrightarrow LV$, where LV is the set of logical values with a distinguished value T. The general definition 15 becomes a following definition for a propositional language $\mathcal{L}$ and its extensional semantics $\mathbf{M}$.

Definition 16 (Sound Propositional Rule under M)

Given a propositional language $\mathcal{L}_{CON}$ and an extensional semantics $\mathbf{M}$*, an inference rule of the form*

$$(r) \qquad \frac{P_1 \;\; ; \;\; P_2 \;\; ; \;\; .... \;\; ; \;\; P_m}{C}$$

is **sound** *under the semantics* $\mathbf{M}$ *if the following condition holds for any $v : VAR \longrightarrow LV$.*

$$\textit{If} \;\; v \models_{\mathbf{M}} \{P_1, P_2, \ldots, P_m\}, \;\; \textit{then} \;\; v \models_{\mathbf{M}} C. \tag{19}$$

Observe that we can rewrite the condition (19) as follow.

$$\text{If } \ v^*(P_1) = v^*(P_2) = \ldots . = v^*(P_m) = T, \ \ \text{then } \ v^*(C) = T. \tag{20}$$

A rule of inference be sound under different semantics, but also rules of inference can be sound under one semantics and not sound under the other.

Example 3 *Given a propositional language $\mathcal{L}_{\{\neg,\cup,\Rightarrow\}}$. Consider two rules of inference:*

$$(r1)\ \frac{(A \Rightarrow B)}{(B \Rightarrow (A \Rightarrow B))} \qquad \textit{and} \qquad (r2)\ \frac{\neg\neg A}{A}.$$

The rule (r1) is sound under classical, **H** *and* **L** *semantics. The (r2) is sound under classical and* **L** *semantics but is not sound under* **H** *semantics.*

Consider the rule (r1).

Let $v : VAR \longrightarrow \{F, T\}$ be any truth assignment, such that $v^*((A \Rightarrow B)) = T$. We use condition (20) and evaluate logical value of the conclusion under v as follows. $v^*((B \Rightarrow (A \Rightarrow B))) = v^*(B) \Rightarrow T = T$ for any formula B and any value of $v^*(B)$. This proves that $v^*(B \Rightarrow (A \Rightarrow B)) = T$ and hence the soundness of (r1). Consider now the **H** semantics. Let $v : VAR \longrightarrow \{F, \perp, T\}$ be any truth assignment, such that $v \models_{\mathbf{M}} (A \Rightarrow B)$, i.e. such that $v^*((A \Rightarrow B)) = T$. We evaluate under **H, L** semantics $v^*((B \Rightarrow (A \Rightarrow B))) = v^*(B) \Rightarrow T$. Now $v^*(B)$ can be T, F as in classical case, or $v^*(B) = \perp$. The case when $v^*(B)$ is T, F is like in classical semantics, so we have to check the case $v^*(B) = \perp$. But in both **H** and **L** semantics $\perp \Rightarrow T = T$. This proves that (r1) is also sound under **H** and **L** semantics.

Consider the rule (r2).

The rule (r2) is sound under classical and **L** by straightforward eveluation. Assume now $v : VAR \longrightarrow \{F, \perp, T\}$ be any truth assignment, such that $v \models_{\mathbf{M}} \neg\neg A$, i.e. such that $v^*(\neg\neg A) = T$ under **H** semantics. We have that $v^*(\neg\neg A) = \neg\neg v^*(A) = T$ if and only if $\neg v^*(A) = F$ if and only if $v^*(A) = T$ or $v^*(A) = \perp$. This proves that that it is possible to have $v \models_{\mathbf{M}} \neg\neg A$ and $v \not\models_{\mathbf{M}} A$, i.e. that (r2) is not sound.

Definition 17 (Strongly Sound Rule under M)

An inference rule $r \in \mathcal{R}$ of the form

$$(r) \qquad \frac{P_1 \ \ ; \ \ P_2 \ \ ; \ \ \ldots . \ \ ; \ \ P_m}{C}$$

is **strongly sound** *under a semantics* **M** *if the following condition holds for all* **M** *models $\mathcal{M}$,*

$$\mathcal{M} \models_{\mathbf{M}} \{P_1, P_2, .P_m\} \quad \textit{if and only if} \quad \mathcal{M} \models_{\mathbf{M}} C. \tag{21}$$

In case of a propositional language $\mathcal{L}_{CON}$ and an extensional semantics $\mathbf{M}$ *the condition (21) is as follows. For for any* $v : VAR \longrightarrow LV$,

$$v \models_{\mathbf{M}} \{P_1, P_2, .P_m\} \quad \textit{if and only if} \quad v \models_{\mathbf{M}} C. \tag{22}$$

We can, and we do state it informally as: " an inference rule $r \in \mathcal{R}$ is strongly sound when the conjunction of all its premisses is logically equivalent under a semantics $\mathbf{M}$ to its conclusion". We denote it informally as

$$P_1 \cap P_2 \cap \ \ldots \ \cap P_m \equiv_{\mathbf{M}} C. \tag{23}$$

Example 4

Given a propositional language $\mathcal{L}_{\{\neg,\cup,\Rightarrow\}}$. *Consider two rules of inference:*

$$(r1)\ \frac{A\ ;\ B}{(A \cup \neg B)} \qquad \textit{and} \qquad (r2)\ \frac{A}{\neg\neg A}.$$

Both rules (r1) and (r2) are sound under classical and $\mathbf{H}$ *semantics. The rule (r2) is strongly sound under classical semantics but is not strongly sound under* $\mathbf{H}$ *semantics. The rule (r1) in not strongly sound under either semantics.*

Consider (r1). Take (in shorthand notation) for $A = T$ and $B = T$. We evaluate $v^*((A \cup \neg B)) = T \cup F = F$ in both semantics. This proves soundness of (r1) under both semantics.. Take now v such that $v(A) = T$ and $v(B) = F$, we get $v^*((A \cup \neg B)) = F \cup T = T$.This proves that $v \models (A \cup \neg B)$ and $v \models_{\mathbf{H}} (A \cup \neg B)$. Obviously $v \not\models \{A, B\}$ and $v \not\models_{\mathbf{H}} \{A, B\}$. this proves that (r1) in not strongly sound under either semantics. Consider (r2). It is strongly sound under classical semantic by (23) and the fact that $A \equiv \neg\neg A$. (r2) is sound under $\mathbf{H}$ semantics. Assume $A = T$. We evaluate (in shorthand notation) $\neg\neg A = \neg\neg T = \neg F = T$. (r2) is not strongly sound under $\mathbf{H}$ semantics. Take v such that $v^*(A) = \perp$, then $v^*(\neg\neg A) = \neg\neg \perp = \neg F = T$. This proves that $A \not\equiv_{\mathbf{H}} \neg\neg A$ and by (23) (r2) is not strongly sound.

Now we are ready to define the notion of a sound and strongly sound proof system. Strongly sound proof systems play a role in constructive proofs of completeness theorem. This is why se introduced and singled them out here.

Definition 18 (Sound Proof System)

Given a proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$.

We say that the proof system S is **sound** *under a semantics* $\mathbf{M}$ *if the following conditions hold.*

1. $LA \subseteq \mathbf{T}_{\mathbf{M}}$; **2.** *Each rule of inference* $r \in \mathcal{R}$ *is* **sound** *under* $\mathbf{M}$.

The proof system S is **strongly sound** *under a semantics* $\mathbf{M}$ *if the condition* **2.** *is replaced by the following condition 2'.*

2'. *Each rule of inference* $r \in \mathcal{R}$ *is* **strongly sound** *under* $\mathbf{M}$.

Example 5 *The proof system S defined below as follows*

$$S = (\mathcal{L}_{\{\neg,\Rightarrow\}},\ \mathcal{F},\ \ \{(\neg\neg A \Rightarrow A),\ (A \Rightarrow (\neg A \Rightarrow B))\},\ \mathcal{R} = \{(r)\})$$

where

$$(r)\ \frac{(A \Rightarrow B)}{(B \Rightarrow (A \Rightarrow B))})$$

is sound, but not strongly sound under classical and **L** *semantics . It is not sound under* **H** *semantics.*

1. Both axioms are basic classical tautologies. Hence to prove that first axiom is **L** tautology we we have to verify only the case (shorthand notation) $A = \perp$. But $\neg\neg \perp \Rightarrow \perp = \neg \perp \Rightarrow \perp = \perp \Rightarrow \perp = T$ and we proved $\models_{\mathbf{L}} (\neg\neg A \Rightarrow A)$. Observe that $(A \Rightarrow (\neg A \Rightarrow B)) = \perp$ if and only if $A = T$ and $(\neg A \Rightarrow B) = \perp$ if and only if $(\neg T \Rightarrow B) = \perp$ if and only if $(F \Rightarrow B) = \perp$, what is impossible undef **L** semantics. Hence $\models_{\mathbf{L}} (A \Rightarrow (\neg A \Rightarrow B))$. We prove, as in example 3, that $\not\models_{\mathbf{H}} (\neg\neg A \Rightarrow A)$, S is not sound under **H** semantics.

2. Obviously $(A \Rightarrow B) \not\equiv (\neg A \Rightarrow B)$, so also $(A \Rightarrow B) \not\equiv_{\mathbf{L}} (\neg A \Rightarrow B)$. This proves by (23) that S in not strongly sound under neither classical not **L** semantic. Nevertheless, it is sound under the both semantics by example 3.

Let $\mathbf{P}_S$ be the set of all provable expressions of S, i.e. $\mathbf{P}_S = \{E \in \mathcal{E} : \ \vdash_S E\}$. Let $\mathbf{T}_{\mathbf{M}}$ be a set of all expressions of S that are tautologies under a semantics $\mathbf{M}$, $\mathbf{T}_{\mathbf{M}} = \{E \in \mathcal{E} : \ \models_{\mathbf{M}} E\}$.

When we define (develop) a proof system S our first goal is to make sure that it a "sound" one, i.e. that all we prove in it is true (with respect to a given semantics). Proving the following theorem establishes this goal.

Theorem 3 (Soundness Theorem)

Given a predicate proof system S and a semantics $\mathbf{M}$.
The following holds.

$$\mathbf{P}_S \subseteq \mathbf{T}_{\mathbf{M}}, \tag{24}$$

i.e. for any $E \in \mathcal{E}$, *the following implication holds*

$$\textit{if } \vdash_S E \quad \textit{then} \quad \models_{\mathbf{M}} E.$$

Proof We prove by Mathematical Induction over the length of a proof that if S is sound as stated in the definition 18, the Soundness Theorem holds for S. It means that in order to prove the Soundness Theorem (under semantics bf M) for a proof system we have to verify the two conditions: **1.** $LA \subseteq \mathbf{T_M}$ and **2.** Each rule of inference $r \in \mathcal{R}$ is **sound** under $\mathbf{M}$.

The next step in developing a logic is to answer next necessary and a difficult question: *Given a proof system S, about which we know that all it proves it true (tautology)with respect to a given semantics. Can we prove all we know to be true (all tautologies) with respect to the given semantics?*

Theorem 4 (Completeness Theorem)

Given a predicate proof system S and a semantics $\mathbf{M}$.
The following holds

$$\mathbf{P}_S = \mathbf{T_M} \tag{25}$$

i.e. for any $E \in \mathcal{E}$, the following holds

$$\vdash_S E \quad \textit{if and only if} \quad \models_{\mathbf{M}} E.$$

The Completeness Theorem consists of two parts:

Part 1: Soundness Theorem: $\mathbf{P}_S \subseteq \mathbf{T_M}$.

Part 2: Completeness Part of the Completeness Theorem: $\mathbf{T_M} \subseteq \mathbf{P}_S$.

Proving the Soundness Theorem for S under a semantics M is usually a straightforward and not a very difficult task. Proving the *Completeness Part* of the Completeness Theorem is always a crucial and very difficult task. There are many methods and techniques for doing so, even for classical proof systems (logics) alone. Non-classical logics often require new sometimes very sophisticated methods. We will study two proofs of the Completeness Theorem for classical propositional Hilbert style proof system in chapter **??**, and a constructive proofs for automated theorem proving systems for classical logic the chapter **??**. We prove provide the proofs of the Completeness Theorem for classical predicate logic in chapter **??** (Hilbert style) and chapter **??**(Gentzen style).

3 Exercises and Homework Problems

Exercise 4

Given a proof system:

$$S = (\mathcal{L}_{\{\neg,\Rightarrow\}},\ \mathcal{E} = \mathcal{F}\ \ LA = \{(A \Rightarrow A), (A \Rightarrow (\neg A \Rightarrow B))\},\ (r)\ \frac{(A \Rightarrow B)}{(B \Rightarrow (A \Rightarrow B))}).$$

1. Prove that S is sound, but not **strongly sound** *under classical semantics.*

2. Prove that S is not sound under $\mathbf{K}$ *semantics.*

3. Write a formal proof in S with 2 applications of the rule (r).

Solution

Parts 1 and 2. In order to prove 1. and 2. we have to verify conditions **1.**, **2.** and bf 2.' of definition 18. Observe that both axioms of S are basic classical tautologies. Consider the rule of inference of S.

$$(r)\ \frac{(A \Rightarrow B)}{(B \Rightarrow (A \Rightarrow B))}.$$

Take any v such that $v^*((A \Rightarrow B))) = T$. We evaluate logical value of the conclusion under the truth assignment v as follows.

$$v^*(B \Rightarrow (A \Rightarrow B)) = v^*(B) \Rightarrow T = T$$

for any B and any value of $v^*(B)$. This proves that S is sound under classical semantics. S is not strongly sound as $(A \Rightarrow B) \not\equiv (B \Rightarrow (A \Rightarrow B))$.

System S is *not sound* under **K** semantics because axiom $(A \Rightarrow A)$ is not a **K** semantics tautology.

Part 3. There are many solutions, i.e. one can construct many forrmal proofs. Here is one of them. For example, one of the formal proofs is a sequence $A_1, A_2, A_3,$ where
$A_1 = (A \Rightarrow A)$
(Axiom)
$A_2 = (A \Rightarrow (A \Rightarrow A))$
Rule (r) application 1 for $A = A,\ B = A$.
$A_3 = ((A \Rightarrow A) \Rightarrow (A \Rightarrow (A \Rightarrow A)))$
Rule (r) application 2 for $A = A, B = (A \Rightarrow A)$.

Exercise 5

Prove, by constructing a formal proof that

$$\vdash_S ((\neg A \Rightarrow B) \Rightarrow (A \Rightarrow (\neg A \Rightarrow B))),$$

where S is the proof system from Exercise 4.

Solution

Required formal proof is a sequence $A_1, A_2,$ where
$A_1 = (A \Rightarrow (\neg A \Rightarrow B))$
Axiom
$A_2 = ((\neg A \Rightarrow B) \Rightarrow (A \Rightarrow (\neg A \Rightarrow B)))$
Rule (r) application for $A = A, B = (\neg A \Rightarrow B)$.

Observe that we needed only one application of the rule (r). One more application of the rule (r) to A_2 gives another solution to Exercise 4, namely a proof A_1, A_2, A_3 for A_1, A_2 defined above and
$A_3 = ((A \Rightarrow (\neg A \Rightarrow B)) \Rightarrow (\neg A \Rightarrow B) \Rightarrow (A \Rightarrow (\neg A \Rightarrow B)))$
Rule (r) application for $A = (\neg A \Rightarrow B)$ and $B = (A \Rightarrow (\neg A \Rightarrow B))$.

Exercise 6

Given a proof system:

$$S = (\mathcal{L}_{\{\cup,\Rightarrow\}}, \quad \mathcal{F}, \quad LA = \{A1, A2\}, \quad (r)\ \frac{(A \Rightarrow B)}{(A \Rightarrow (A \Rightarrow B))}\ ),$$

where $A1 = (A \Rightarrow (A \cup B))$, $\quad A2 = (A \Rightarrow (B \Rightarrow A))$.

1. Prove that S is sound under classical semantics and determine whether S it is sound or not sound under **K** *semantics.*

2. Write a formal proof B_1, B_2, B_3 in S with 2 applications of the rule (r) that starts with axiom A1, i.e such that $B_1 = A1$.

3. Write a formal proof B_1, B_2 in S with 1 application of the rule (r) that starts with axiom A2, i.e such that $A_1 = A2$.

Solution

Part 1. Axioms of S are basic classical tautologies. The proof (in shorthand notation) of soundness of the rule of inference is the following. Assume $(A \Rightarrow B) = T$. Hence the logical value of conclusion is $(A \Rightarrow (A \Rightarrow B)) = (A \Rightarrow T) = T$ for all A. S is *not sound* under **K** semantics. Let's take truth assignment such that $A = \perp, B = \perp$. The logical value of axiom A1 is as follows.
$(A \Rightarrow (A \cup B)) = (\perp \Rightarrow (\perp \cup \perp)) = \perp$ and $\not\models_{\mathbf{K}} (A \Rightarrow (A \cup B))$. Observe that the v such that $A = \perp, B = \perp$ is not the only v that makes $A1 \neq T$, i.e. proves that $\not\models_{\mathbf{K}}$ A1.
$(A \Rightarrow (A \cup B)) \neq T$ if and only if $(A \Rightarrow (A \cup B)) = F$ or $(A \Rightarrow (A \cup B)) = \perp$. The first case is impossible because $A1$ is a classical tautology. Consider the second case. $(A \Rightarrow (A \cup B)) = \perp$ in two cases. c1. $A = \perp$ and $(A \cup B) = F$, i.e. $(\perp \cup B) = F$, what is impossible. c2. $A = T$ and $(A \cup B) = \perp$, i.e. $(T \cup B) = \perp$, what is impossible. c3. $A = \perp$ and $(A \cup B) = \perp$, i.e. $(\perp \cup B) = \perp$. This is possible for $B = \perp$ or $B = F$, i.e when $A = \perp, B = \perp$ or $A = \perp, B = F$. From the above observation we get a *second solution.* S is not sound under **K** semantics. Axiom A1 is not **K** semantics tautology. There are exactly two truth assignments v, such that $v \not\models$ A1. One is, as defined in the first solution, namely $A = \perp, B = \perp$. The second is $A = \perp, B = F$.

Part 2. The formal proof B_1, B_2, B_3 is as follows.
$B_1 = (A \Rightarrow (A \cup B))$
Axiom
$B_2 = (A \Rightarrow (A \Rightarrow (A \cup B)))$
Rule (r) application for $A = A$ and $B = (A \cup B)$
$B_3 = (A \Rightarrow (A \Rightarrow (A \Rightarrow (A \cup B))))$
Rule (r) application for $A = A$ and $B = (A \Rightarrow (A \cup B))$.

Part 3. The formal proof B_1, B_2 is as follows.
$B_1 = (A \Rightarrow (B \Rightarrow A))$
Axiom

$B_2 = (A \Rightarrow (A \Rightarrow (B \Rightarrow A)))$.
Rule (r) application for $A = A$ and $B = (B \Rightarrow A)$.

Exercise 7

Let S be the following proof system:

$$S = (\ \mathcal{L}_{\{\Rightarrow, \cup, \neg\}},\quad \mathcal{F},\quad A1,\quad (r1),\ (r2)\ ),$$

where the logical axiom A1 is: $A1 = (A \Rightarrow (A \cup B))$,

Rules of inference (r1), (r2) are:

$$(r1)\ \frac{A\ ;B}{(A \cup \neg B)},\qquad (r2)\ \frac{A\ ;(A \cup B)}{B}.$$

1. *Verify whether S is sound/not sound under classical semantics.*
2. *Find a formal proof of* $\neg(A \Rightarrow (A \cup B))$ *in S, ie. show that*
$$\vdash_S \neg(A \Rightarrow (A \cup B)).$$
3. *Does* $\vdash_S \neg(A \Rightarrow (A \cup B))$ *prove that* $\models \neg(A \Rightarrow (A \cup B))$*?*

Solution
Part 1. The system is not sound. Take any v such that $v^*(A) = T$ and $v^*(B) = F$. The premiss $(A \cup B$ of the rule $(r2)$ is T under v, and the conclusion under v is $v^*(B) = F$.

Part 2. The proof of $\neg(A \Rightarrow (A \cup B))$ is as follows.
B_1: $(A \Rightarrow (A \cup B))$,
axiom
B_2: $(A \Rightarrow (A \cup B))$,
axiom
B_3: $((A \Rightarrow (A \cup B)) \cup \neg(A \Rightarrow (A \cup B)))$,
rule (r1) application to B_1 and B_2
B_4: $\neg(A \Rightarrow (A \cup B))$,
rule (r2) application to B_1 and B_3).

Part 3. System S is not sound, so existence of a proof does not guarantee that what we proved is a tautology. Moreover, the proof of $\neg(A \Rightarrow (A \cup B))$ used rule $(r2)$ that is not sound.

Exercise 8

Create a 3 valued extensional semantics **M** *for the language* $\mathcal{L}_{\{\neg,\ \mathbf{L},\ \cup,\ \Rightarrow\}}$ *by defining the connectives* $\neg$, $\cup$, $\Rightarrow$ *on a set* $\{F, \perp, T\}$ *of logical values. You must follow the following* **assumptions a1, a2**.

a1 *The third logical value value is intermediate between truth and falsity, i.e. the set of logical values is ordered as follows:* $F < \perp < T$.
a2 *The value T is the designated value. The semantics has to model a situation in which one "likes" only truth; i.e. the connective* $\mathbf{L}$ *must be such that* $\mathbf{L}T = T$, $\mathbf{L}\perp = F$, *and* $\mathbf{L}F = F$. *The connectives* $\neg$, $\cup$, $\Rightarrow$ *can be defined as you wish, but you have to define them in such a way to make sure that*

$$\models_{\mathbf{M}} (\mathbf{L}A \cup \neg \mathbf{L}A).$$

Solution
Here is a simple **M** semantics. We define the logical connectives by writing functions defining connectives in form of the truth tables.

M Semantics

L	F	$\perp$	T
	F	F	T

$\neg$	F	$\perp$	T
	T	F	F

$\cap$	F	$\perp$	T
F	F	F	F
$\perp$	F	$\perp$	$\perp$
T	F	$\perp$	T

$\cup$	F	$\perp$	T
F	F	$\perp$	T
$\perp$	$\perp$	T	T
T	T	T	T

$\Rightarrow$	F	$\perp$	T
F	T	T	T
$\perp$	T	$\perp$	T
T	F	F	T

We verify whether the condition **s3** is satisfied, i.e. whether $\models_{\mathbf{LK}} (LA \cup \neg LA)$ by simple evaluation. Let $v : VAR \longrightarrow \{F, \perp, T\}$ be any truth assignment. For any formula A, $v^*(A) \in \{F, \perp, T\}$ and $LF \cup \neg LF = LF \cup \neg LF = F \cup \neg F \cup T = T$, $L \perp \cup \neg L \perp = F \cup \neg F = F \cup T = T$, $LT \cup \neg LT = T \cup \neg T = F \cup T = T$.

We verify whether $\models_{\mathbf{M}} (LA \cup \neg LA)$ by simple evaluation. Let $v : VAR \longrightarrow \{F, \perp, T\}$ be any truth assignment. For any formula A, $v^*(A) \in \{F, \perp, T\}$ and $LF \cup \neg LF = LF \cup \neg LF = F \cup \neg F \cup T = T$, $L \perp \cup \neg L \perp = F \cup \neg F = F \cup T = T$, $LT \cup \neg LT = T \cup \neg T = F \cup T = T$.

Exercise 9

Let S be the following proof system

$$S = (\ \mathcal{L}_{\{\neg, \mathbf{L}, \cup, \Rightarrow\}},\ \mathcal{F},\ \ \{A1, A2\},\ \ \{(r1),\ (r2)\}\ )$$

where the logical axioms A1, A2 and rules of inference (r1), (r2) defined for any formulas $A, B \in \mathcal{F}$ *as follows.*

A1 $(\mathbf{L}A \cup \neg \mathbf{L}A)$,
A2 $(A \Rightarrow \mathbf{L}A)$,

$$(r1)\ \frac{A\ ;B}{(A \cup B)}, \qquad\qquad (r2)\ \frac{A}{\mathbf{L}(A \Rightarrow B)}.$$

1. Show, by constructing a proper formal proof that

$$\vdash_S ((\mathbf{L}b \cup \neg\mathbf{L}b) \cup \mathbf{L}((\mathbf{L}a \cup \neg\mathbf{L}a) \Rightarrow b))).$$

Please, write comments how each step of the proof was obtained

2. Verify whether the system S is **M***-sound funder the semantics* **M** *you have developed in Exercise 8. You can use shorthand notation.*

3. If the system S **is not sound/ sound** *under your semantics* **M** *then redefine the connectives in a way that such obtained new semantics* **N** *would make S* **sound/not sound**

Solution
Part 1. Here is the formal proof $B_1, B_2,\ B_3,\ B_4$.

B_1: $(\mathbf{L}a \cup \neg\mathbf{L}a)$, axiom $A1$ for $A = a$,
B_2: $\mathbf{L}((\mathbf{L}a \cup \neg\mathbf{L}a) \Rightarrow b)$, rule (r2) for $B = b$ applied to B_1,
B_3: $(\mathbf{L}b \cup \neg\mathbf{L}Ab)$, axiom $A1$ for A$= b$,
B_4: $((\mathbf{L}b \cup \neg\mathbf{L}b) \cup \mathbf{L}((\mathbf{L}a \cup \neg\mathbf{L}a) \Rightarrow b))$, rule (r1) applied to B_3 and B_2.

Part 2. Observe that both logical axioms of S are **M** tautologies. A1 is **M** tautology by definition of the semantics, A2 is **M** tautology by direct evaluation. Rule (r1) is **sound** because when $A = T$ and $B = T$ we get $A \cup B = T \cup T = T$. Rule (r2) is **not sound** because when $A = T$ and $B = F$ (or $B = \perp$) we get $\mathbf{L}(A \Rightarrow B) = \mathbf{L}(T \Rightarrow F) = \mathbf{L}F = F$ (or $\mathbf{L}(T \Rightarrow \perp) = \mathbf{L} \perp = F$).

Part 3. In order to make the rule $(r2)$ sound while preserving the soundness of axioms A1, A2 we have to modify only the definition of implication. Here is the **N** semantics implication

$\Rightarrow$	F	$\perp$	T
F	T	T	T
$\perp$	T	$\perp$	T
T	T	T	T

Observe that it would be hard to convince anybody to use our sound proof system S, as it would be hard to convince anybody to adopt our **N** semantics.

Homework Problems

1. Given a proof system $S = (\mathcal{L}_{\{\cup,\Rightarrow\}},\ \mathcal{F},\ LA = \{A1, A2\},\ (r)\ \frac{(A\Rightarrow B)}{(A\Rightarrow(A\Rightarrow B))}\ )$, where $A1 = (A \Rightarrow (A \cup B)),\ A2 = (A \Rightarrow (B \Rightarrow A))$. Prove, by constructing a formal proof in S that $\vdash_S (A \Rightarrow (A \Rightarrow (A \Rightarrow (A \Rightarrow A))))$.

   Does it prove that $\models (A \Rightarrow (A \Rightarrow (A \Rightarrow (A \Rightarrow A))))$.

2. Given a proof system: $S = (\mathcal{L}_{\{\neg,\Rightarrow\}}, \quad \mathcal{E} = \mathcal{F} \quad LA = \{(A \Rightarrow A), (A \Rightarrow (\neg A \Rightarrow B))\}, \quad (r)\ \frac{(A \Rightarrow B)}{(B \Rightarrow (A \Rightarrow B))})$.

   (i) Prove that S is sound under classical semantics.

   (ii) Prove that S is not sound under **K** semantics.

   (iii) Write a formal proof in S with 3 applications of the rule (r).

   (iv) Prove, by constructing a formal proof that $\vdash_S ((\neg A \Rightarrow B) \Rightarrow (A \Rightarrow (\neg A \Rightarrow B)))$.

3. Given a proof system: $S = (\mathcal{L}_{\{\cup,\Rightarrow\}}, \quad \mathcal{E} = \mathcal{F} \quad LA = \{A1, A2\}, \quad \mathcal{R} = \{(r)\}$), where $A1 = (A \Rightarrow (A \cup B))$, $A2 = (A \Rightarrow (B \Rightarrow A))$ and $(r)\ \frac{(A \Rightarrow B)}{(A \Rightarrow (A \Rightarrow B))}$.

   (i) Prove that S is *sound* under classical semantics.

   (ii) Determine whether S is *sound* or *not sound* under **L** semantics.

   (iii) Write a formal proof in S with 3 applications of the rule (r) that starts with axiom A1.

   (iv) Does it prove/ that $\models_{\mathbf{L}} A$ for a formula A obtained in (iii).

   (v) Prove, by constructing a formal proof in S that $\vdash_S (A \Rightarrow (A \Rightarrow (A \Rightarrow (A \Rightarrow A))))$. Does it prove (and why) that $\models (A \Rightarrow (A \Rightarrow (A \Rightarrow (A \Rightarrow A))))$.

4. S is the following proof system: $S = (\ \mathcal{L}_{\{\Rightarrow,\cup,\neg\}}, \quad \mathcal{F}, \quad LA = \{(A \Rightarrow (A \cup B))\} \ \ (r1),\ (r2)$), where $(r1)\ \frac{A\ ;B}{(A \cup \neg B)}$, $(r2)\ \frac{A\ ;(A \cup B)}{B}$.

   (i) Verify whether S is sound/not sound under classical semantics. (ii) Find a formal proof of $\neg(A \Rightarrow (A \cup B))$ in S, i. e. show that $\vdash_S \neg(A \Rightarrow (A \cup B))$.

   (iii) Does above (ii) prove that $\models \neg(A \Rightarrow (A \cup B))$?

5. By a m-valued semantics S_m, for a propositional language $\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ we understand any definition of of connectives $\neg, \cap, \cup, \Rightarrow$ as operations on a set $L_m = \{l_1, l_2, ...l_m\}$ of logical values (for $m \geq 2$).

   We assume that $l_1 \leq l_2 \leq ... \leq l_m$, i.e. L_m is totally ordered by a certain relation $\leq$ with l_1, l_m being smallest and greatest elements, respectively. We denote $l_1 = F$, $l_m = T$ and call them (total) False and Truth, respectively. For example, when $m = 2$, $L_2 = \{F, T\}$, $F \leq T$. Semantics S_2 is called a classical semantics if the connectives are defined as $x \cup y = max\{x, y\}$, $x \cap y = min\{x, y\}$, $\neg T = F, \neg F = T$, and $x \Rightarrow y = \neg x \cup y$, for any $x, y \in L_2$.

   Let VAR be a set of propositional variables of $\mathcal{L}$ and let S_m be any m-valued semantics for $\mathcal{L}$. A truth assignment $v : VAR \longrightarrow L_m$ is called a S_m model for a formula A of $\mathcal{L}$ if and only if $v^*(A) = T$ and logical value $v^*(A)$ is evaluated accordingly to the semantics S_m. We denote is symbolically as $v \models_{S_m} A$.

Let $S = (\mathcal{L},\ \mathcal{F},\ \{A1, A2, A3\},\quad MP\ \frac{A\ ;(A \Rightarrow B)}{B})$ be a proof system with logical axioms:

A1: $(A \Rightarrow (B \Rightarrow A))$,

A2: $((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)))$,

A3: $((\neg B \Rightarrow \neg A) \Rightarrow ((\neg B \Rightarrow A) \Rightarrow B))$.

(i) Prove that S is sound under S_2 classical semantics.

(ii) Define your own S_2 semantics under which S is not sound.

(iii) Define your own S_4 semantics under which S is sound and other S_4 semantics under which S is not sound.

(iv) Define your own S_n semantics such that S is sound for all for $2 \leq n \leq m$.

(v) Show, by construction a formal proof, that $\vdash_S (A \Rightarrow A)$.