CHAPTER 7

GENERAL PROOF SYSTEMS

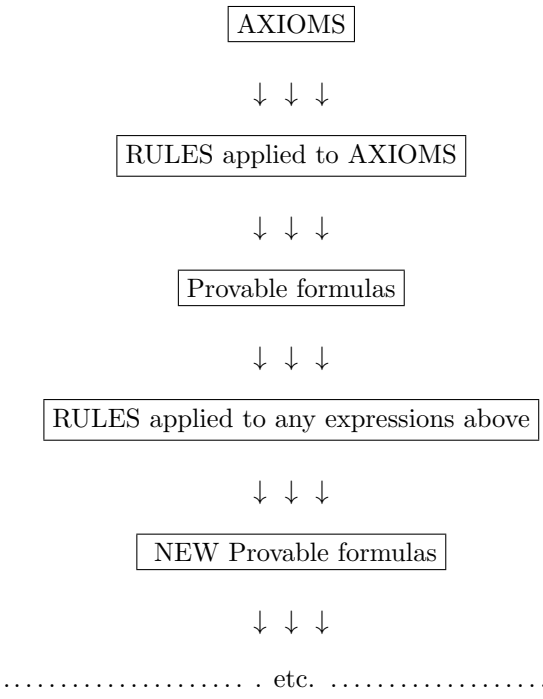
1 Introduction

Proof systems are built to prove statements. They can be thought as an *inference machine* with special statements, called *provable statements*, or sometimes *theorems* being its final products. The starting points are called *axioms of the system*. We distinguish two kinds of axioms: *logical*LA and *specific* SX.

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 logical axioms LA is also called *a logic proof system*.

If we build a proof system for which there is no known semantics, like 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*.

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* and can be depicted as follows:



The provable formulas for which we have a formal proof are called *consequences* of the axioms, or theorem, or just simple *provable formulas*. 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, i.e. generating from axioms via the rules. Rules with this property are called *sound rules* and the system *a sound proof system*. The theorem establishing the soundness of a given proof system is called *Soundness Theorem*. It states in a case of a logic proof system S that for any formula A of the language of the system S, A is provable in a (logic) proof system S, then A is a tautology.

A proof system S with logical axioms LA and specific axioms SA is called *a formal theory* with specific axioms SA, based on a logic defined by the axioms LA. We denote a formal theory by $\mathcal{TH}_S(SA)$, or $\mathcal{TH}(SA)$ when the proof system S is fixed.

In a case of *a formal theory* $\mathcal{TH}(SA)$ the *Soundness Theorem* says: for any formula A of the language of the theory $\mathcal{TH}(SA)$, if a formula A is provable in the theory $\mathcal{TH}(SA)$, then A is true in any model of the set of specific axioms SA.

Any 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 Ł logic semantics, or K logic semantics, or others. This is why we talk about proof systems for classical logic, for modal logic, for intuitionistic logic, etc. In general there are many proof systems that are sound under a given semantics, i.e. many sound proof systems for a given logic. We present some examples at the end of the chapter.

Given a logic 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 *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 following theorem.

Completeness Theorem (for a logic system S, under a semantics M. For any A (of the language of S)),

A is provable in S if and only if A is a tautology under the semantics M.

We write it symbolically as:

$$\vdash_S A \quad iff \quad \models_M A.$$

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

Proving the *completeness part* of the **Completeness Theorem** is always a very difficult, and a crucial task. We will study two proofs of the Completeness Theorem for classical propositional Hilbert style proof system in the next chapter, and a constructive proofs for automated theorem proving systems for classical logic the the following chapter.

Observe that we formulated all these basic theorems linking semantics and syntax (provability) in a general manner. In this part of the book we consider only propositional languages, and hence **Completeness Theorems** for propositional logics as examples. The case of Predicate Logics will be discussed in the second part of the book.

2 Proof Systems and Proofs

In this section we formulate a general definition of a proof system and all its components. We also define the 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, the language $\mathcal{L}$ we work with. It can be a propositional language, but it can be any other formal language.

Component: Language $\mathcal{L}$ of S

Usually, as in the propositional case, the language $\mathcal{L}$ consists of its alphabet $\mathcal{A}$ and a set of formulas, denoted here by $\mathcal{F}$, i.e.

$$\mathcal{L} = (\mathcal{A}, \mathcal{F}).$$

We assume that the both sets $\mathcal{A}$ and $\mathcal{F}$ are enumerable, i.e. we will deal here with enumerable languages only.

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.

For example, as we will see later, we might consider the set of all finite sequences of formulas, or sets of formulas, or other expressions, called Gentzen sequents, or sets of clauses in the case of the resolution based systems as the basic *expressions* of our proof system S under consideration.

Component: Expressions $\mathcal{E}$

Given a language $\mathcal{L}$. We assume that $\mathcal{E}$ is enumerable and primitively recursive set built out of the alphabet of $\mathcal{L}$ and some additional symbols, if needed. I.e. we assume that there is an effective procedure to determine whether a given expression is, or is not in $\mathcal{E}$.

The expressions are often built out of the formulas $\mathcal{F}$ of the language $\mathcal{L}$ and some, if needed, extra symbols. In many proof system we choose the set of formulas $\mathcal{F}$ as expressions, i.e. $\mathcal{F} = \mathcal{E}$.

Semantical Link We always have to *extend our semantics* of the language $\mathcal{L}$ (if given) from the set of formulas, to the set of expressions. I.e. we have to define, for any expression $E \in \mathcal{E}$ what does it mean that E is a tautology under the semantics of $\mathcal{L}$.

We often do it by establishing a semantic equivalency 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 $E \equiv_M A$.

Example 1

In the automated theorem proving system **RS** we study in chapter 10, our basic 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. $\Delta \equiv (A_1 \cup A_2 \cup ... \cup A_n)$. This means that the sequence

$$\Delta = A_1, A_2, ..A_n$$

which is an expression of $\mathcal{E}$ is semantically equivalent to the formula

$$A = (A_1 \cup A_2 \cup ... \cup A_n)$$

The proof system acts as an inference machine, with provable formulas being its final products. This inference machine is defined by first setting, as a starting point a certain non-empty, proper subset AX of $\mathcal{F}$, called a set of axioms of the system S.

Component: Axioms Given a non- empty set $\mathcal{E}$ of expressions of a language $\mathcal{L}$. Any proper, non-empty, primitively recursive subset AX of the set $\mathcal{E}$ is called a **set of axioms** of S. I.e. there is an effective procedure to determine whether a given expression $A \in \mathcal{E}$ is in AX or not.

Semantical link : For a given semantics M for $\mathcal{L}$ and its extension to $\mathcal{E}$, we usually choose as AX a subset of expressions that are **tautologies** under the semantics M.

Component: Rules of Inference

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 the proof system investigations are often called *syntactic methods* as opposed to semantic methods, which deal with the semantics of the language only. The semantical connection does exists and is established by Soundness and Completeness theorems and will be discussed in detail later.

We assume that a proof system contains only **a finite number** of inference rules. The set of rules of inference is denoted by $\mathcal{R}$.

We assume additionally that each rule has a finite number of premisses and one conclusion, i.e. is defined by a certain relation, which assigns a single expression, called conclusion, to a finite string of expressions, called its premisses.

Moreover, we also assume that one can effectively decide, for any rule, whether a given string of expressions form its premisses and conclusion or not, i.e. that all relations $r \in \mathcal{R}$ are primitively recursive.

More formally, we say that:

(1) for each $r \in \mathcal{R}$, there is a number m, called the *number of premisses* of r, such that r **is a relation** defined in $\mathcal{E}^m$ with values in $\mathcal{E}$, i.e.

$$r \subseteq \mathcal{E}^m \times \mathcal{E}$$

(2) all $r \in \mathcal{R}$ are primitively recursive relations.

We put it in a formal definition as follows.

Definition 2.1 (Rule of Inference) *Given a non- empty set $\mathcal{E}$ of expressions of a language $\mathcal{L}$ of S.*

Any primitively recursive relation

$$r \subseteq \mathcal{E}^m \times \mathcal{E}, \quad m \geq 1$$

is called **a rule of inference**.

For any $(P_1, ..., P_m, A) \in r$, $P_1, ..., P_m$ are called the **premisses**, A is called the **conclusion**, and m the **number** of premisses of r.

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

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

$$(r) \qquad \frac{A}{B},$$

where A is a premiss, and B is a conclusion of r.

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

$$(r) \qquad \frac{P_1 \ ; \ P_2}{A},$$

where P_1, P_2 are the premisses of r and A is its conclusion. P_1 is called a left premiss of r and P_2 is called a right premiss.

If r is a three premisses rule and $(P_1, P_2, P_3, A) \in r$, then we write it accordingly as

$$(r) \quad \frac{P_1 \ ; \ P_2 \ ; \ P_3}{A},$$

where P_1, P_2, P_3 are the premisses of r and A is its conclusion.

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

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

where $P_1, P_2, ..., P_m$ are the premisses of r and A is its conclusion.

Semantical link: For a given semantic M for $\mathcal{L}$, and $\mathcal{E}$, we chose for S rules which preserve truthfulness under the semantics M, i.e. we prove that the rules of the system S are *sound*.

Now we are ready to define formally a proof system.

Definition 2.2 (Proof system S) *By a proof system we understand a triple*

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

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* $\mathcal{L}$*,*
- AX *is a set of axioms of the system. It is a non-empty, proper, primitively recursive subset of the set of expressions* $\mathcal{E}$*,*
- $\mathcal{R}$ *is a finite set of rules of inference (as defined in definition 2.1).*

Finitely Axiomatizable Systems When the set AX of axioms of S is **finite** we say that the system S has a *finite axiomatization*, or is *finitely axiomatizable.*

Infinitely Axiomatizable Systems are system that are not finitely axiomatizable, i.e. systems with infinitely many axioms.

In our book we consider only finitely axiomatizable systems.

Provable expressions of a system S are final products of a single or multiple use of the inference rules, with axioms taken as a starting point. A single use of an inference rule is called a direct consequence, a multiple application of rules of inference is called a proof. Formal definitions are as follows.

Direct consequence

A conclusion of a rule of inference is called a direct consequence of its premisses. I.e. for any rule of inference r, if

$$(P_1, ...P_n, A) \in r,$$

then A is called a direct consequence of $P_1, ...P_n$ by virtue of r.

Proof of A in S

A proof of an expression $A \in \mathcal{E}$ in a proof system S is a sequence

$$A_1, ...A_n$$

of expressions from $\mathcal{E}$, such that,

$$A_1 \ \in AX, \quad A_n = A$$

and for each i, $1 \leq i \leq n$, either A_i is an axiom of the system or A_i is a direct consequence of some of the preceding expressions by virtue of one of the rules of inference.

We write

$$\vdash_S A$$

to denote that A has a proof in S. When the proof system S is fixed we write

$$\vdash \ A.$$

Provable expressions of S

Any expression A such that A has a proof in S, i.e.

$$\vdash_{\mathcal{S}} A$$

is called a provable expression of S.

The set of all provable expressions of S is denoted by $\mathbf{P}_S$, i.e.

$$\mathbf{P}_S = \{A \in \mathcal{E} : \quad \vdash_{\mathcal{S}} \mathcal{A}\}.$$

While proving expressions, we often need to use some extra information available, besides the axioms of the proof system. To describe formally this process we introduce the notions of *a proof from a given set of expressions.*

Proof of A from Γ in S

Given a set Γ of expressions of a proof system, by a proof of A from Γ we will understand a proof where the expressions from Γ are added to the set LA of the axioms of the system.

Definition 2.3 *A proof of A from Γ is a sequence*

$$A_1, ...A_n$$

of expressions, such that
(i) *$A_n = A$, and*
(ii) *for each i, $1 \leq i \leq n$, either A_i is an axiom of the system, or*
(iii) *A_i is in Γ, or*
(iii) *A_i is a direct consequence of some of the preceding expressions by virtue of one of the rules of inference.*

We write

$$\Gamma \vdash_S A$$

to denote that A has a proof from Γ in S and

$$\Gamma \vdash A,$$

when the system S is fixed.

The set of all expressions provable from Γ (and logical axioms LA in S is denoted by $\mathbf{P}_S(\Gamma)$, i.e.

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

Hypothesis

If $\Gamma \vdash_S A$ then the expressions of Γ are called hypotheses of the proof of A from Γ.

Finite Γ

If Γ is a finite set and $\Gamma = \{B_1, B_2, ..., B_n\}$, then we write

$$B_1, B_2, ..., B_n \vdash_{\mathcal{S}} A$$

instead of $\{B_1, B_2, ..., B_n\} \vdash_{\mathcal{S}} A$.

Empty Γ

The case of $\Gamma = \emptyset$ is a special one. By the definition of a proof of A from Γ, $\emptyset \vdash A$ means that in the proof of A only axiomsLA of S were used. We hence write

$$\vdash_S A$$

to denote that A has a proof from empty Γ.

Consequence of Γ

If $\Gamma \vdash_S A$ then A is called a consequence of Γ in S.

Consequence operation in S is a function $\mathbf{Cn}_S$ which to every set $\Gamma \subseteq \mathcal{E}$, assigns a set of all its consequences. I.e.

$$\mathbf{Cn}_S : 2^{\mathcal{E}} \longrightarrow 2^{\mathcal{E}}$$

such that for every $\Gamma \in 2^{\mathcal{E}}$

$$\mathbf{Cn}_S(\Gamma) = \{A \in \mathcal{E} : \ \Gamma \vdash_S A\}.$$

The consequence operation $\mathbf{Cn}_S$ has the following properties.

Monotonicity

For any sets Γ, Δ of expressions of S,

if $\Gamma \subset \Delta$ then $\mathbf{Cn}_S(\Gamma) \subseteq \mathbf{Cn}_S(\Delta)$.

The monotonicity is often expressed as a fact that adding new information to the system do not invalidate old provable formulas.

Transitivity

For any sets $\Gamma_1, \Gamma_2, \Gamma_3$ of expressions of S,

if $\Gamma_1 \subseteq \mathbf{Cn}_S(\Gamma_2)$ and $\Gamma_2 \subseteq \mathbf{Cn}_S(\Gamma_3)$, then $\Gamma_1 \subseteq \mathbf{Cn}_S(\Gamma_3)$.

The transitivity property is often stated in the following way: if A is a consequence of a set Δ, such that all elements of Δ are consequences of Γ, then A is also a consequence of Γ.

Finiteness

For any expression $A \in \mathcal{F}$ and any set $\Gamma \subseteq \mathcal{F}$,

$A \in \mathbf{Cn}_S(\Gamma)$ if and only if there is a finite subset Γ_0 of Γ such that $A \in \mathbf{Cn}_S(\Gamma_0)$.

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.

Decidable system A proof system, for which there is a mechanical method for determining, given any expression A of the system, whether there is a proof of A, is called *a decidable proof system*, otherwise it is called *undecidable*.

Observe that the above notion of decidability of the system does not require to find a proof, it requires only a mechanical procedure of deciding whether *there is* a proof of any expression of the system.

Example 1

A Hilbert style proof system for classical propositional logic is an example of a decidable, but not syntactically, or automatically decidable proof system. We conclude its decidability from the Completeness Theorem and the decidability of the notion of classical tautology.

The proof system such that a mechanical procedure of finding proofs for its expressions will be called syntactically decidable, or an automated proof system.

Syntactically decidable system A proof system S, for which there is a mechanical method for determining, given any expression A of the system, not only whether there is a proof of A, but which also generates a proof, is called *syntactically decidable* or *automatically decidable*, or *an automated system*; otherwise S is not not syntactically decidable.

Example 2

The Gentzen proof system, the **RS** system presented here later, and Resolution style proof systems for classical propositional logic are the examples of both decidable and syntactically decidable proof systems, or *automated proof systems*.

3 Some Examples

The notion of a proof in a system S usually carries a semantical meaning via the *Soundness Theorem* 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 A of a system one has to construct of s sequence of proper transformations, defined by the rules of inference.

Example 1

System S_1

Let S_1 be the following proof system:

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

where A, B are any formulas of the propositional language $\mathcal{L}_{\{P,\Rightarrow\}}$.

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

The following system S_2, even if it looks very much alike the system S_1, will produce, as we will see, a different set of provable formulas.

Example 2

System S_2

Let S_2 be the following proof system:

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

where $a \in VAR$ and B is any formulas of the propositional language $\mathcal{L}_{\{P,\Rightarrow\}}$.

Observe that for example a formula $((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ is of the form $A \Rightarrow A$ for A being $(Pa \Rightarrow (b \Rightarrow c))$, and hence is an *axiom* of S_1, 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.

Here are some examples of provable formulas and formal proofs in both systems.

1. Of course,
$$\vdash_{S_1} ((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$$
and the axiom
$$((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$$
forms a formal proof of itself in S_1.

2. The formulas $P(a \Rightarrow a)$, and $PP(a \Rightarrow a)$ are provable formulas of both, S_1 and S_2, i.e.

$$\vdash P(a \Rightarrow a),$$

$$\vdash PP(a \Rightarrow a$$

for S_1 and S_2.

The formal proofs in both systems of above formulas are identical and are as follows.

The formal proof of $P(a \Rightarrow a)$ in S_1 and S_2 is:

$(a \Rightarrow a)$,	$P(a \Rightarrow a)$.
axiom	rule application for $B = (a \Rightarrow a)$

The formal proof of $PP(a \Rightarrow a)$ in S_1 and S_2 is:

$(a \Rightarrow a)$,	$P(a \Rightarrow a)$,	$PP(a \Rightarrow a)$.
axiom	rule application for $B = (a \Rightarrow a)$	rule application for $B = P(a \Rightarrow a)$

3. The formula $PPP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ is a theorem of S_1 i.e.

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

but it is not a theorem of S_2, i.e.

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

The formal proof of $PPP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$ in S_2 is:

$((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$,	$P((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$,
axiom	rule r application
$PP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$,	$PPP((Pa \Rightarrow (b \Rightarrow c)) \Rightarrow (Pa \Rightarrow (b \Rightarrow c)))$.
rule r application	rule r application

Let's now search for a proof of our formula in S_2. If it had the proof, the only last step in this proof would be 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 be 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 procedure is an effective, automatic procedure 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.

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 proves the following.

Fact Proof systems S_1 and S_2 are syntactically decidable.

Observe also, that the systems S_1 and S_2 are such that we can easily describe the general form of their provable formulas, namely we have the following.

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

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

where P^n denotes n-iteration of P.

Obviously we have that $PR_{S_1} \neq PR_{S_2}$, and $PR_{S_2} \subseteq PR_{S_1}$.

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

Semantical link : We haven't defined a semantics for the language $\mathcal{L}_{\{\Rightarrow,P\}}$ of systems $S1, S2$, so we can't talk about the soundness of these systems yet. But all known modal semantics *extend the classical semantics*,i.e. are the same as the classical one on non-modal connectives so the axiom in both cases would be a sound axiom. Hence to assure the soundness of both systems we must have a modal semantics M such that the rule

$$(r)\ \frac{B}{PB}$$

is sound, i.e. such that

$$\models_M (B \Rightarrow PB).$$

Otherwise they will not be sound.

Example 3

Systems S_3 and S_4

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;

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

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

1. Describe the sets of provable formulas of S_3 and S_4.
2. Do they produce the same sets of provable formulas? I.e. is it *true/ false* that
$$\{A : \vdash_{S_3} A\} = \{A : \vdash_{S_4} A\}.$$
If yes, prove it, if not give a formula which is a theorem of one system and is not a theorem of other system.
3. Are the systems S_3 and S_4 decidable?

Let's first describe the set of provable formulas of both systems.

System S_3 Obviously, $\vdash_{S_3}(A \cup \neg A)$. One application of the inference rule gives us the proof of $((A \cup \neg A) \cup (A \cup \neg A))$. The next application of the rule (to the already produced formula) will give us the proof of $(((A \cup \neg A) \cup (A \cup \neg A)) \cup (A \cup \neg A))$. It is easy to see that all provable formulas of S_3 will be of the form of the disjunction of the axiom of S_3, what we denote as follows:

$$PR_{S_3} = \{\bigcup_n (A \cup \neg A)^n : n \in N, A \in \mathcal{F}\},$$

where $\bigcup_n (A \cup \neg A)^n$ denotes a disjunction of n formulas of the form $(A \cup \neg A)$.

System S_4 Obviously, as before, $\vdash_{S_4}(A \cup \neg A)$. One application of the inference rule gives us the proof of $(B \cup (A \cup \neg A))$, where B is a certain formula from $\mathcal{F}$. The rule can't be, by its definition, applied to already produced theorem (except for a theorem which is the axiom), so the multiple application of the rule means application to the axiom and producing all possible formulas of the form $(B \cup (A \cup \neg A))$, where B's are different for different applications. Hence

$$PR_{S_4} = \{(B \cup (A \cup \neg A)) : A, B \in \mathcal{F}\} \cup \{(A \cup \neg A) : A \in \mathcal{F}\}.$$

Provable formulas of S_3 and S_4 Obviously, $PR_{S_3} \subseteq PR_{S_4}$, as we have, by definition, that $\bigcup_n (A \cup \neg A)^n = \bigcup_{n-1}(A \cup \neg A)^{n-1} \cup (A \cup \neg A)$ and $\bigcup_{n-1}(A \cup \neg A)^{n-1}$ is a formula of $\mathcal{L}_{\{\cup,\neg\}}$. We can denote it by B. I.e. we have proved that any theorem of S_3 is an axiom or has a form $(B \cup (A \cup \neg A))$ for a certain $B \in \mathcal{F}$.

The formula $((a \cup \neg b) \cup (a \cup \neg a))$ has a proof in S_4, i.e.

$$\vdash_{S_4}((a \cup \neg b) \cup (a \cup \neg a)),$$

but obviously

$$\nvdash_{S_3}((a \cup \neg b) \cup (a \cup \neg a)).$$

The above proves that

$$PR_{S_3} \subseteq PR_{S_4} \quad and \quad PR_{S_3} \neq PR_{S_4}.$$

Syntactically Decidable It follows immediately form the form of the sets PR_{S_3} and PR_{S_4} that both systems are syntactically decidable. The design of the proper proof searching procedure is left to the reader as an exercise.

Semantical link 1 : Both systems are *sound under classical semantics.*

It follows from the fact that

$$\models (A \cup \neg A),$$

and the rules

$$\frac{B}{(B \cup (A \cup \neg A))}, \qquad \frac{(A \cup \neg A)}{(B \cup (A \cup \neg A))}$$

are sound because

$$\models (B \Rightarrow (A \cup B)).$$

Semantical link 2 : Both systems are *not sound under* **Ł, K, H, B** *semantics.*

It follows from the fact that

$$\not\models_M (A \cup \neg A)$$

for $M = Ł, \mathbf{K}, \mathbf{H}, \mathbf{B}$.

The following natural questions arise.

Q1 Are all proof systems syntactically decidable?

Q2 Can we give an example of a proof system for a given logic which is decidable and not syntactically decidable, but the logic does have (another) syntactically decidable system?

The answers are straightforward: no, not all the systems are syntactically decidable, or even decidable. Moreover, as we will see, the most "natural" and historically first developed proof systems for classical propositional logic (it means was a complete proof system with respect to classical semantics) is not syntactically decidable. Any proof system for classical predicate logic is not decidable. There are however, in a case of propositional classical logic syntactically systems. The proof of their decidability, undecidability, syntactical decidability will be presented later. They are neither obvious nor straightforward.

4 Soundness and Completeness

Proof systems always have some semantical links. Sometimes they are developed for a language with a given semantics, sometimes they are developed in order to axiomatically characterize an unknown yet semantics. In first case the future goal is to prove how a known semantics relates to the proof system. In the second case the goal is first to develop a semantics, and then to prove how it relates to the proof system. The relationship between proof system S and its semantics is always stated in form of two theorems: Soundness Theorem and Completeness Theorem.

5 Exercises and Homework Problems

Exercise 1

Given a proof system:

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

1. Prove that S is *sound* under classical semantics.

2. Prove that S is *not sound* under **K** semantics.

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

Solution of 1.

Definition: System S is sound if and only if

(i) Axioms are tautologies and

(ii) rules of inference are sound, i.e lead from all true premisses to a true conclusion.

We verify the conditions (i), (ii) of the definition as follows.

(i) Both axioms of S are basic classical tautologies.

(ii) Consider the rule of inference of S.

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

Assume that its premise (the only premise) is True, i.e. let v be any truth assignment, 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 $v^*(B)$.

Solution of 2. System S is *not sound* under **K** semantics because axiom $(A \Rightarrow A)$ is not a **K** semantics tautology.

Solution of 3. There are many solutions. Here is one of them.

Required formal proof 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 2

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 1.

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 1, 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 3

Given a proof system:

$$S = (\mathcal{L}_{\{\cup,\Rightarrow\}},\ \ \mathcal{E} = \mathcal{F}\ \ AX = \{A1, A2\},\ \ \mathcal{R} = \{(r)\}\ ),$$

where

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

and

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

Prove that S is *sound* under classical semantics.

Solution: Axioms of S are basic classical tautologies. The proof 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.

Exercise 4

Determine whether S from the Exercise 3 is *sound* or *not sound* under **K** semantics.

Solution 1: 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 second solution.

Solution 2: 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 Solution 1: $A = \perp, B = \perp$. The second is $A = \perp, B = F$.

Exercise 5

Write a formal proof A_1, A_2, A_3 in S from the Exercise 3 with 2 applications of the rule (r) that starts with axiom A1, i.e such that $A_1 = A1$.

Solution: The formal proof A_1, A_2, A_3 is as follows.
$A_1 = (A \Rightarrow (A \cup B))$
Axiom
$A_2 = (A \Rightarrow (A \Rightarrow (A \cup B)))$
Rule (r) application for $A = A$ and $B = (A \cup B)$

$A_3 = (A \Rightarrow (A \Rightarrow (A \Rightarrow (A \cup B))))$
Rule (r) application for $A = A$ and $B = (A \Rightarrow (A \cup B))$.

Exercise 6

Use results from Exercise 4 to determine whether $\models_{\mathbf{K}} A_3$.

Solution 1: We use the two v from QUESTION 3 to evaluate the logical value of A_3. Namely we evaluate: $v^*(A \Rightarrow (A \Rightarrow (A \Rightarrow (A \cup B)))) = (\bot \Rightarrow (\bot \Rightarrow (\bot \Rightarrow (\bot \cup \bot)))) = \bot$, or $v^*(A \Rightarrow (A \Rightarrow (A \Rightarrow (A \cup B)))) = (\bot \Rightarrow (\bot \Rightarrow (\bot \Rightarrow (\bot \cup F)))) = \bot$. Both cases prove that $\not\models_{\mathbf{K}} A_3$.

Solution 2: We know that S is not sound, because there is v for which $A1 = A_1 = \bot$, as evaluated in Exercise 4. We prove that the rule (r) preserves the logical value $\bot$ under any v such that $A1 = \bot$. as follows.
Let the premiss $(A \Rightarrow B) = \bot$, the logical value of the conclusion is hence $(A \Rightarrow \bot) = \bot$ for $A = \bot, T$ and $(A \Rightarrow \bot) = T$ for $A = F$.
The case $A = F$ evaluates the premiss $(A \Rightarrow B) = (F \Rightarrow B) = T$ for all B, what contradicts the assumption that $(A \Rightarrow B) = \bot$. We must hence have $A = \bot$. But all possible v for which $A1 = \bot$ are such that $A = \bot$, what end the proof.
It means that any A such that A has proof that starts with axiom $A1$ and then multiple applications of the rule (r) is evaluated to $\bot$ under all v, such that $v^*(A1) = \bot$. Hence, in particular, $\not\models_{\mathbf{K}} A_3$.

Exercise 7

Write a formal proof A_1, A_2 in S from the Exercise 3 with 1 application of the rule (r) that starts with axiom A2, i.e such that $A_1 = A2$.

Solution: The formal proof A_1, A_2 is as follows.
$A_2 = (A \Rightarrow (B \Rightarrow A))$
Axiom
$A_2 = (A \Rightarrow (A \Rightarrow (B \Rightarrow A)))$
Rule (r) application for $A = A$ and $B = (B \Rightarrow A)$.

Exercise 8

Use results from Exercise 3 to determine whether $\models A_2$.

Solution: System S is sound under classical semantics, hence by the Soundness Theorem we get that $\models (A \Rightarrow (A \Rightarrow (B \Rightarrow A)))$, as it has a proof in S.

Exercise 9

Prove, by constructing a formal proof in S from the Exercise 3 that

$$\vdash_S (A \Rightarrow (A \Rightarrow (A \Rightarrow (A \Rightarrow A)))).$$

Solution: $A_2 = (A \Rightarrow (A \Rightarrow A))$
Axiom for $B = A$
$A_2 = (A \Rightarrow (A \Rightarrow (A \Rightarrow A)))$
Rule (r) application for $A = A$ and $B = (A \Rightarrow A)$.
$(A \Rightarrow (A \Rightarrow (A \Rightarrow (A \Rightarrow A))))$
Rule (r) application for $A = A$ and $B = (A \Rightarrow (A \Rightarrow A))$.