# Learning Deductive Reasoning

**MSc Thesis** *(Afstudeerscriptie)*

written by

**Ana Lucia Vargas Sandoval**
(born March 27th, 1989 in Morelia, Mexico)

under the supervision of **Dr Nina Gierasimczuk** and **Dr Jakub Szymanik**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

**MSc in Logic**

at the *Universiteit van Amsterdam.*

| **Date of the public defense:** | **Members of the Thesis Committee:** |
|---|---|
| *August 24, 2015* | Prof. Dr Pieter Adriaans |
| | Dr Theo Janssen |
| | Prof. Dr Dick de Jongh |
| | Prof. Dr Ronald de Wolf |
| | Dr Jelle Zuidema |

Institute for Logic, Language and Computation

# Contents

# Acknowledgements

Firstly, I wish to express my sincere gratitude to my advisors for their continuous help and guidance. To Dr Nina Gierasimczuk for always encouraging my personal motivations and ideas, for providing me with new challenges, and for inspiring me to not be afraid on taking risks in scientific research. To Dr Jakub Szymanik for trusting me with my choice of research on a topic completely unknown for me initially; for his sincere advice, patience, and for indicating the importance of setting boundaries in my (sometimes chaotic) exploration. To my student mentor Prof. Dick de Jongh for his infinite patience, guidance, and immense knowledge; for his always objective advice and for showing me the importance of skepticism and critique in scientific research. Their guidance helped me in all the time of research and writing of this thesis. I can not imagine having better advisors for my master study.

I would like to thank the members of the committee for their interest in the research I conducted, and for their comprehension and support on my restricted graduating deadline. When it came to discussing my work or anything even remotely related, it was very interesting to talk with Dr Jelle Zuidema (ILLC), Prof. James W. Garson (University of Houston) and many others, whose observations and contributions are present in my research.

I would like to thank all members of faculty and staff, associates, and friends from the Institute of Logic, Language and Computation for creating such an amazing scientific environment, for encouraging the development of original ideas, and for the great social environment. In particular, to my gang: Eileen, Pastan, Nigel, Sarah, Kostis, KH, Iliana, Shimpei, and specially to my almost brother Aldo, for his gentle advice and always optimistic perspective about *all the things one does in life*.

To my closest family: to my father for pushing my love for science and desire for knowledge and wisdom my entire life. To my mom for guiding me with freedom and love into an spiritual life, showing me, since I was little, the power of faith and kindness, and for encouraging me to follow my dreams. To my older sister and younger brother for their support, love, and advice.

# Abstract

In this thesis we study the phenomenon of *learning deductive reasoning* by presenting a formal learning theory model for a class of possible proof systems which are built by misinterpretations of the rules in natural deduction system of classical logic. We will address this learning problem with an abstract computational procedure representation at the level of formulas and proofs. That said, we can point out that the main goal for our model is to propose a learner who: (1) is able to effectively learn a deductive system, and (2) within the learning process, the learner is expected to disambiguate, i.e., choose one deductive system over other possibilities. With these goals in mind, we evaluate and analyze different methods of presenting data to a learning function. One of the main observations is that the way in which information is presented; by means of positive data only or by means of mixed data with teacher's intervention, plays a crucial role in the learning process.

# Chapter 1

# Introduction

Imagine that your family from far away is coming to visit you. A long time has passed since you last saw them, so you wish to festively welcome them. Thus, you decide to make a dinner reservation in a nice restaurant on the day of their arrival.

Now the day has come, you pick them up at the airport. You chat on your way to the restaurant. At some point your uncle says that during the flight he read a magazine article about dreams and their connection with human cognitive abilities. Your uncle explains what the article said:

– *It was very interesting! The article said that some novel scientific results suggest that if a man is smart, then he dreams a lot while sleeping; and he usually remembers his dreams with clarity on the next day.*

Already having some doubts concerning the reliability of such magazine articles, you hear your aunt saying to your uncle:

– *Don't you dream often?*

And then to you:

– *Your uncle always shares his dreams with me the next morning, actually he gives very detail descriptions of his dreams.*

From that your uncle concludes, laughing:

– *You are right! So, according to the article, I'm a smart person!*

Then you think to yourself: *Well... not really*. But why is that you are skeptical about the conclusion your uncle just made? Your concerns are not about your uncle being smart or not, they are more about if he can really conclude that from the given premises.

After some time, when you are at the restaurant trying to decide between the roasted chicken and the lasagna, you hear your uncle ordering the caramelized duck. Then the following exchange takes place between him and the waiter:

– *Which side dish would you like with the duck, sir?*

– *Well... here it says that it comes with steam rice or fried vegetables, right?*

– *Precisely sir, you need to choose which one you want.*

– *Mmmmm... but I don't understand. Isn't it the case that the duck can be served with both? That is what is written here in the menu!*

A little bit puzzled, the waiter explains pointing at the words in the menu:

– *Well sir, what this* means *is that you can either choose rice or vegetables.*

Not very convinced, your uncle replies pointing at the word "or" in the menu:

– *But this* means *that I can actually have both, doesn't it?*

the waiter replies impatiently:

– *I'm sorry sir, but you will have to choose only one of these two options.*

to what your uncle ends up saying, rolling his eyes:

– *Whatever, I'll just get the rice then.*

What happened here? It is not that your uncle is an irrational man or that he was playing a fool. He just has a different interpretation for the conditional ($IF \dots THEN$) and for disjunction ($OR$). Judging from his utterances, we could say that your uncle is "reversing" the direction of the conditional; and that he considers disjunction as an *inclusive* disjunction, so he interprets $OR$ more as a conjunction. Why is

it that your uncle acquired such interpretations? What kind of inferential system corresponds to such interpretations? After all, they seem quite plausible as alternatives for the usual interpretation of logical connectives. But what if your uncle were someone that when seeing a sentence of the form $A \wedge B$, he infer $\neg A$? This should be considered as a possibility, as weird as it may sound, one-in-a-million peculiar case.

As a matter of fact misinterpretations of this kind arise more often than one would imagine. In this thesis we will address this phenomenon from the perspective of the possibility of *learning* alternative inference systems. Those alternatives often arise from misinterpretations of logical connectives. Our motivation for studying such phenomenon comes from empirical research showing a large gap between the normative accounts of logical reasoning and human performance on logical tasks. Experiments with logical reasoning show that there are patterns in errors made by subjects, the mistakes are often not random. For instance, in (1 - Gierasimczuk et al.), the authors propose a way of analyzing logical reasoning focusing on a deductive version of the Mastermind game for children. The observed patterns in the erroneous reasoning seem to suggest that they could be caused by misinterpreting some logical connectives. In another line of reasoning, (2 - Fugard et al.) investigated how subjects interpret conditionals in a probability-logical setting. It has been observed that truth conditions might not play a significant role in human reasoning, leading to new evidence against the material interpretation of conditional statements.

These patterns seem more visible when studying learning progress of participants on a certain task, which leads psychologists to analyze the learning process via a sequence of cognitive strategies that are adopted and later given up for better ones. All these studies seem to agree on the importance of distinguishing errors that stem from initial misunderstanding of the premises from those that steam from later parts of the deductive process (for instance a misapplication of logical rules). Moreover, their results seem to suggest that the meaning of logical connectives is initially obscured, allowing participants to assign to them any possible meaning. This phenomenon, we believe, can lead people to acquire alternative inference systems.

When thinking about human learning and human reasoning some natural questions arise:

- *What does it mean to learn? Can we model any learning process? How can it be defined computationally?* Learning theorists address these questions using mathematical and computational techniques. Their general formalizations bring us closer to robust answers with potentially useful applications.

- *What does it mean to reason? Can we model the process of acquiring deductive reasoning?* Psychologists, cognitive scientists, and philosophers have investigated these questions in many different paradigms. However, it seems that there is no consensus on the basic mechanism behind human reasoning.

The notion of *learning* has been addressed from many different angles, empirical studies in psychology to machine learning and formal areas within computer science. The notion of *reasoning* and *rationality* has been addressed in areas like philosophy, logic, mathematics, and even economics; where the normativity of classical logic seems to chase away any attempt to formalize natural logical reasoning. Can we push to go far beyond the idea of "*the* logic we should all follow", to explore the phenomenon of how and why it is that people can possess *different* reasoning schemes? Can such such reasoning schemes can be *acquired*, and if so, by what means?

To address this issue, we make use of two major paradigms in mathematical logic and computer science: proof theory and formal learning theory. The former is used for representing and analyzing proofs as formal mathematical objects; typically presented as inductively-defined data structures such as lists or trees, which are constructed according to the axioms and rules of inference of the logical system (3 - Buss). In this thesis we will focus our attention on natural deduction proof system for propositional logic, developed independently by mathematicians Jaśkowski (1929) and Gentzen (1934) in an attempt to characterize the *real practice of proving* in mathematics. Formal learning theory, on the other hand, gives a computational framework for investigating the process of conjecture change (4 - Jain et al.), concerned with the global process of convergence in terms of computability. Our research will be based on Gold's framework of identification in the limit (5 - Gold), which provides direct implications for the analysis of grammar inference and language acquisition (6 - Angluin and Smith) and scientific discovery (7 - Kelly).

Why did we choose these precise areas for treating the problem? The simple answer is that the goals of these theories and our investigations are aligned:

**Formal learning theory** has been conceived as an attempt to formalize and understand the process of language acquisition. In accordance with his nativist theory of language acquisition and his mathematical approach to linguistics, Chomsky (1965) proposed the existence of what he called a *language acquisition device*, a module that humans are born with, in order to learn language. Later on, this turned out to be only a step away from the formal definition of language learners as functions in Gold's work, that on infinitely large finite samples of language keep outputting conjectures (supposedly grammars) which correspond to the language in question. In an analogy to a child, who on the basis of finite samples learns to creatively use a language, by inferring an appropriate set of rules, learning functions are supposed to stabilize on the value that encodes a finite set of rules for generating the language. The generalization of this concept in the context of computability theory has taken the learners to be number-theoretic functions that on finite samples of a recursive set output indices that encode Turing machines, in an attempt to find an index of a machine that generates the set.

**Proof theory** arises with the goal of analyzing the main features of mathematical proofs. The first accounts for this were based on axiomatic systems as in the Hilbert tradition, however there were several opponents of this view and a general discomfort with these systems as mathematicians do not seem to construct their proofs by means of an axiomatic theory. This was Jaśkowski's and Gentzen's point of departure for the design of a formal system capturing a more realistic process of mathematical reasoning, which Gentzen called *natural deduction*. The system is modular in nature, it contains a reasoning module for each logical connective, without the need for defining one connective in terms of another. Nowadays, many mathematicians and logicians have declared it to be the most *intuitive* deductive system; and it has even been considered by psychologists and cognitive scientists to build their theories concerning human reasoning, as is the case of (8 - Rips) and (9 - Braine and O'Brien).

In this thesis, we initiate the study of the problem of *learning deductive reasoning*. Is it the case that we are born with some sort of *logical device* allowing us to reason (in the spirit of Chomsky's Universal Grammar)? Could this logical device be something similar to the *natural deduction* system in proof theory (given its arguably "intuitive" nature)? Or is it the case that we *learn* to *perform* these kinds of *valid* reasonings by learning the right, or appropriate proof system? We will partially address these questions by presenting a formal learning theory model for a class of possible proof systems which are built by misinterpretations of the rules in natural deduction system. We will address this learning problem with an abstract computational procedure representation at the level of formulas and proofs. That said, we can point out that the main goal for our model is to propose a learner who: (1) is able to effectively learn a deductive system; and (2) within the learning process, the learner is expected to disambiguate (i.e., choose one deductive system over other possibilities) on the basis of reasoning patterns he observes. With these goals in mind, we evaluate and analyze different methods of presenting data to a learning function. *Our analysis suggest that there may be basic, intrinsic parts of the deductive-reasoning mechanism in humans (a type of structural inferential system is given as the starting point); and there are other parts which need to be* learned *by means of presenting adequate information (a system corresponding, e.g., to the adequate interpretation of connectives)*. One of the main observations is that the way in which information is presented; by means of positive data only and mixed data with teacher intervention, plays a crucial role in the learning procedure.

The content of this thesis is organized in two parts. Let us give a brief overview of their corresponding chapters.

Part I (Chapters 2 and 3) is concerned with introducing the concepts, tools and results from formal learning theory and the natural deduction proof system that will be significant for our study. Chapter 2 is dedicated to formal learning theory. We present the basic notions, terminology and best known results. We focus on Gold's concept of learning, *identification in the limit*. We end this chapter with a discussion concerning the relevance of formal learning theory for cognitive science. Chapter 3 is dedicated to the natural deduction proof system for propositional logic. In the first two sections we present natural deduction in the simplest way; being as close as possible to Gentzen's terminology. In Section 3.3 we address several alternative ways of representing the inference, in particular for the natural deduction system; and the possible implications each representation may carry. We will focus on three possible ways of representation: 1) as a grammar, where the language is conceived as the set of complete proofs that the inference system produces by using propositional formulas; 2) as an axiomatic system; and 3)

as scheme of rules operating as rules for reasoning. We will evaluate the advantages and disadvantages in each case; concluding that a hybrid of these forms will be the "ideal" representation.

Part II (Chapters 4 to 5) is concerned with the description of our learning model and the obtained results. Chapter 4 will be dedicated to the mathematical formalization of the alternative inference systems and the construction of the learning space. First, in Section 4.1, we will define natural deduction system in new terms, where each rule will be given as classes of functions. We will also provide a corresponding notion of proof and, in Section 4.2, we show it's correspondence to the usual natural deduction system. In Section 4.3 we will define the possible misinterpretations of natural deduction rules, which later will constitute a class of possibilities the learner can choose as inference systems from the learning space. In Chapter 5 we evaluate five different methods of presenting the data to a learning function which corresponds to different environments, having different implications for the learning process. We conclude that the last method requires the intervention of a teacher for easier disambiguation between alternatives. We formalize and implement this idea by supervising the learning procedure with an adequate teacher for the class of alternative systems.

Chapter 6 concludes the thesis by giving an overview of results, possible extensions of our model, and suggestions for future work.

# Chapter 2

# Learning theory

## 2.1 Introduction

Formal Learning Theory deals with the question of how an agent should use observations about her environment to arrive at correct and informative conclusions. Philosophers have developed learning theory as a normative framework for scientific reasoning and inductive inference. The basic set-up of learning frameworks is as follows. We have a collection of inputs and outputs, and an unknown relationship between the two. We do have a class of hypotheses describing this relationship, and suppose one of them is correct (the hypothesis class can be either finite or infinite). A learning algorithm takes in a set of inputs, the data and produces a hypothesis for these data. Generally we assume the data are generated by some random process, and the hypothesis changes as the data change. The main idea behind a learning model in these terms is that: if we supply enough data, we can converge to a hypothesis which is *accurate* for the data.

In this chapter the formal concepts and terminology of learning theory are presented. We will discuss the origins and history of this field in the following section. Then, formal definitions and known results will be presented. Since we will only be interested, for our framework, in Gold's concept of learning *identification in the limit*; it will be explained in more detail in Section 2.3. Finally in Section 2.4 we will discuss the relevance of the collaboration with cognitive science and its implications in the field.

### 2.1.1 History

In the attempt to formalize the philosophical notion of *inductive inference* and building a computational approach for studying *language acquisition*; formal learning theory emerged encompassing and succeeding in addressing these two problems. The entire field stems from five remarkable papers:

1. (10 - Solomonoff) developed a Bayesian inference approach, nowadays considered the statistical inference learning model. It was originally conceived as a theory of universal inductive inference; thus a theory of prediction based on logical observations in which prediction is done using a completely Bayesian framework. In short, a theory for predicting the next symbol in a countable source basing this prediction from a given series of symbols. The only assumption that the theory makes is that there is an unknown but computable probability distribution for presenting the data. It is a mathematical formalization of Occam's razor (11) and the Principle of Multiple Explanations (12).

2. (5 - Gold) gave a recursion theoretic approach in terms of classes of *recursively enumerable languages* (subsets of the natural numbers). Among other facts, Gold demonstrated that no procedure guarantees success in stabilizing to an arbitrarily chosen finite-state grammar on the basis of a presentation of strings of the language generated by the grammar. In particular, Gold showed that no procedure is successful on a collection that includes an infinite language and all of its finite subsets. He introduced a learning framework called *identification in the limit*. His results revealed the relevance of formal learning theory to the acquisition of language by infants. The idea had its origins in one of the firsts attempts in using mathematical methods on linguistics. Chomsky,

the pioneer on this field, proposed the existence of what he called *language acquisition device,* an innate module humans poses, in order to acquire language.

3. (13 - Putnam) introduced the idea of a computable procedure for converting data into conjectures about a hidden recursive function (the data are increasing initial segments of the function's graph). He proved the non-existence of computable procedures that guarantee success, his results contrasted with the goals for inductive logic announced by Carnap (1950).

4. (14 - Blum and Blum) introduced novel techniques to prove unexpected theorems about paradigms close to Putnam's and Gold's. Among their discoveries is the surprising fact that there is a collection $F$ of total recursive 0-1 valued functions such that a computable procedure can achieve Gold-style success on $F$, but no computable procedure can successfully estimate (beyond the original observation range) the value of a variable on the basis of its relationship with another variable of all functions in $F$.

5. (15 - Valiant) introduced a new framework in learning theory called *Probably Approximately Correct* or PAC learning which birthed a new sub-field of computer science called *computational learning theory.* In this framework probability theory gets involved in a way that changes the basic nature of the learning problem used in previous learning models. PAC was developed to explain how effective behavior can be learned. The model shows that pragmatically coping with a problem can provide a satisfactory solution in the absence of any theory of the problem. Valiant's theory exposes the shared computational nature of learning and evolution, showing some light on longstanding questions such as nature versus nurture; and the limits of humans and artificial intelligence.

We can say that Learning theory has been originally designed as an attempt to formalize and understand the process of language acquisition, but it has widened its scope in the last few decades. Researchers in machine learning tackled related problems (the most famous being that of inferring a deterministic finite automaton, given examples and counter-examples of strings). There have been several important extensions of the recursion theoretic approach of Gold in the field, for instance the notion of tell-tale sets introduced by (16 - Angluin). She also gave the notion of *active learning* in her work on identification with the help of more powerful clues (17), like membership queries and equivalence queries (18). An important negative result is given by (19 - Pitt and Warmuth) in which by complexity inspired results, they expose the hardness of different learning problems. Similarly following Valiant's framework, from computational linguistics, one can point out the different systems introduced to automatically build grammars from sentences (20, 21). In more applied areas, such as speech recognition, visual recognition and even computational biology, researchers also worked on learning grammars or automata from strings (see, e.g., 22). Reviews of related work in specific fields can be found in (23, 24, 25).

### 2.1.2 How does it work?

In contrast to other philosophical approaches of inductive inference, formal learning theory does not aim to describe a universal inductive method or explicate general rules of inductive rationality. Rather, learning theory pursues a context-dependent means-ends analysis: *For a given empirical problem and a set of cognitive goals, what is the best method for achieving the goals?* Most of learning theory examines which investigative strategies reliably and efficiently lead to correct beliefs about the world.

Learning theory, seen with the eyes of computer science, is concerned with the process of convergence in terms of computability, i.e. with sequences of outputs of recursive functions, with special attention for those functions that get settled on an appropriate value (5, 10, 13). The goal is to address the possibility of inferring coherent conclusions from partial, step wise given information. The learners are functions, in special cases the learners are recursive functions. If the learners are recursive, there are some cases in which full certainty can be achieved in a computable way. Thus the learner obtains full certainty when the objective ambiguity between alternatives disappears. Several types of learners can be studied and how they make use of the given information. In order to study the phenomenon of reaching certainty in a more efficient way, a new agent denoted as *teacher* can be introduced.

An important thing to point out about *learning* in Learning theory is the fact that when an agent $A$ "learned that $\phi$", this means something more than to declare to have learned something. The incoming information is vital and it is spread over more than one single step in the inductive process. The step-by-step nature of this inference is important since the incoming data are of a different nature than the

thing being learned. Usually, the "teacher" (environment, nature, etc) gives only partial information about a set. Thus the relationship between data and hypotheses is like the one between sentences and grammars, natural numbers and Turing machines, derivations and proof systems. If we are aware of the hypothesis, we can infer the type of possible data that is going to occur, but in principle we will not be able to make a conclusive inference from data to hypotheses. Thus, we say that an agent $A$ "learned that a hypothesis holds" if he converged to this hypothesis because of data that are consistent with the actual world.

Some questions arise naturally: What is it about an empirical question that allows inquiry to reliably arrive at the correct answer? What general insights can we gain into how reliable methods go about testing hypotheses? Learning theorists answer these questions with characterization theorems, generally of the form "it is possible to attain this standard of empirical success in a given inductive problem if and only if the inductive problem meets the following conditions". Characterization theorems tell us how the structure of reliable methods corresponds to the structure of the hypotheses under investigation. The characterization result draws a line between solvable and unsolvable problems. Background knowledge reduces the inductive complexity of a problem; with enough background knowledge, the problem crosses the threshold between the unsolvable and the solvable. In many domains of empirical inquiry, the pivotal background assumptions are those that make reliable inquiry feasible.

## 2.2 Basic definitions

In principle, learning theory (in the broader sense) can be described for any situation and classes of objects. To provide insights of its powerful usage, just for now we will focus on the situation of learning sets of integers. The possibilities (sets of integers) will be often called languages. Sometimes we will also view learning theory in terms of language acquisition so that the possibilities will be grammars.

Let $\mathcal{U} \subseteq \mathbb{N}$ be an infinite recursive set; we call any $S \subseteq U$ a language.

**Definition 1** *A* language learnability model *will be composed by the following elements:*

1. *A class of* concepts *that needs to be learned.*
2. *A definition of learnability: establishes the requirements to claim that something has being learned.*
3. *A method of information: the "format" in which information will be presented to the learner.*
4. *A naming relation which assigns names to languages (perhaps more that one). The names are understood as grammars.*

In the general case, computational learning theory is interested in indexed families of recursive languages, i.e., classes $\mathcal{C}$ for which a computable function $f : U \times \mathbb{N} \to \{0, 1\}$ exists that uniformly decides $\mathcal{C}$. Formally[1]

$$f(x, i) = \begin{cases} 1 & \text{if } x \in S_i \\ \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

The class under *learning* consideration $\mathcal{C}$ can be finite or infinite. We will often refer to the class $\mathcal{C}$ containing the possible hypothesis or alternatives as the *learning space*. The input for the learner is given as an infinite stream of data $\epsilon$. The method of presenting information to the learner can be either of positive elements only which are elements that correspond to the language that is being learned (often called *texts*); or containing some negative elements which are elements that do not correspond to the target language. When we have positive and negative data in the stream, we say that the method of presenting $\epsilon$ is *informative*; often called *informative teacher*. The examples provided in this chapter will consider only positive streams of data.

[1] This is the approach to identification in the limit due to (16).

**Definition 2** *By a* positive stream of data $\epsilon$ *of* $S \in \mathcal{C}$ *we mean an infinite sequence of elements from* $S$ *enumerating all and only the elements from* $S$ *allowing repetitions.*

**Definition 3** *To simplify things we will use the following notation:*

1. $\epsilon$ *will denote an infinite sequence of data. In this sense* $\epsilon$ *is a countable stream of clues;*
2. $\epsilon_n$ *is the n-th element of* $\epsilon$*;*
3. $\epsilon \upharpoonright n$ *is the sequence* $(\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$*;*
4. $set(\epsilon)$ is the set of elements that occur in $\epsilon$;
5. Let $U^*$ be the set of all finite sequences over a set $U$. If $\alpha, \beta \in U^*$, then by $\alpha \sqsubset \beta$ we mean a $\alpha$ is a proper initial segment of $\beta$;

**Definition 4** *A* learning function $L$ *is a recursive map from finite data sequences to indexes of hypotheses,* $L : U^* \to I_{\mathcal{C}}$*, where* $I_{\mathcal{C}}$ *is an index set for the learning space* $\mathcal{C}$ *under consideration.*

The *learner* identifies a language by stating one of its names, i.e., one of its grammars.

Sometimes the function will be allowed to refrain from giving an index number answer in which the output is marked by $\uparrow$. In this context of learning functions, symbol $\uparrow$ should not be read as a calculation that does not stop.

We can think of formal learning theory as a collection of theorems and claims about games of the following character:

- Players: A learner ($A$) and a teacher ($T$).
- Game pieces: A class $\mathcal{C}$ of elements of any nature. This corresponds to the possible learning space. An infinite stream of data. This corresponds to the pieces of information related to one or many elements in $\mathcal{C}$.
- Goal: This varies, from learning one particular element of the class to learning the whole class. In the first simple form, the teacher selects *a priori* some $S^* \in \mathcal{C}$ to be the target to learn.
- Goal of the learner: To name the actual hypothesis, i.e., the one the teacher selected.

Rather than present Formal Learning Theory in further detail, we rely on the examples given below to communicate its flavor. They illustrate the fundamental factors of essentially all paradigms embraced by the theory.

**Example 1** *Guessing a numerical set: Consider two agents* $A$ *and* $T$ *playing a clue game. The goal of player* $T$ *is to choose a subset of the natural numbers which is hard for player* $A$ *to guess. Clearly the goal of player* $A$ *is to guess* $T$*'s set choice. The rules of the game are the following:*

1. *Both players agree on a family* $\mathcal{C}$ *of non empty sets of the natural numbers* $\mathbb{N}$ *that are legal choices.*
2. *Player* $T$ *chooses an* $S \in \mathcal{C}$ *denoted by* $S^T$ *and an infinite countable list* $\epsilon$ *consisting in all and only the elements in* $S^T$*. Each element* $\epsilon_k$ *of this list is a clue for* $A$ *to help him come up with the correct set.*
3. *Player* $T$ *will provide to* $A$ *the elements in* $\epsilon$ *step-by-step .*
4. *After a finite number of clues provided by* $T$ *to* $A$*, player* $A$ *needs to declare his guess about the identity of* $S^T$*.*
5. *If player* $A$ *choice is accurate, he wins. Otherwise* $T$ *wins.*

*Now let us play. Assume* $\mathcal{C}_1 = \{S_i = \{0, i\} : i \in \mathbb{N} \setminus 0\}$ *is the class players* $A$ *and* $T$ *agreed on. Suppose player* $T$ *chooses* $S^T = \{0, 154\}$ *and that the infinite list that will provide the clues is* $\epsilon = \{0, 0, \ldots, 0, 154, 0, \ldots, 0, \ldots\}$ *such that* $\epsilon_{68} = 154$ *which means that the* $68^{th}$ *member of the list if the number* $154 \in S^T$ *and the rest of the members of the list are number* $0$*. All things considered,* $T$ *starts giving* $A$ *the clues:*

*Starting step:* $\epsilon_0 = 0$,

*Consecutive step:* $\epsilon_1 = 0$,

.
.
.

$67^{th}$ *step:* $\epsilon_{67} = 0$,

$68^{th}$ *step:* $\epsilon_{68} = 154$,

*After the* $68^{th}$ *step in the game, player* $A$ *announces that he wants to make a guess. Since he knows the nature of the sets in* $\mathcal{C}$*, he can easily infer that* $T$*'s choice was* $S_{154}$*. Thus in this instance of the game* $A$ *won.*

It seems that for class $\mathcal{C} = \{S_i = \{0, i\} : i \in \mathbb{N} \setminus 0\}$, player $A$ has a huge advantage over player $T$ cause there are not sets which are hard enough to guess. Moreover, for this class player $A$ always wins.

Now let us use the example above but for a more interesting class of sets.

**Example 2** *Assume* $\mathcal{C}_2 = \{S_i = \mathbb{N} \setminus \{i\} : i \in \mathbb{N}\}$*, this is the class of subsets in* $\mathbb{N}$ *that are missing exactly one number. Suppose* $T$ *chooses* $S^T = \mathbb{N} \setminus \{3\}$ *and a list* $\epsilon$ *for elements in* $S^T$*. Class* $\mathcal{C}_2$ *is harder to learn, so one guess it is not enough for player* $A$ *to have chances of winning the game. Therefore, in this game, player* $A$ *is allowed to make more than one guess. Actually* $A$ *is allowed to make a guess after each clue. Player* $A$ *wins only if after finitely many guesses, he will continue to guess the hypothesis corresponding to* $\epsilon$*. Now* $T$ *starts giving clues:*

*Starting step:* $\epsilon_0 = 4$*, player* $A$ *makes a guess which is not* $S_4$*;*

*Consecutive step:* $\epsilon_1 = 50$*, player* $A$ *makes a guess which is not in* $\{S_4, S_{50}\}$

*2nd step:* $\epsilon_2 = 7$*, player* $A$ *makes a guess which is not in* $\{S_4, S_{50}, S_7\}$*;*

.
.
.

*the process continues.*

*This game never stops and it seems that player* $A$ *does not have any chance to win. However* $A$ *does have chances to win if he uses an effective procedure to make his guesses. If at each stage player* $A$ *guesses* $\mathbb{N} \setminus \{i_0\}$ *where* $i_0$ *the least number not yet revealed by* $T$*, by using this procedure player* $A$ *has a strategy which will make him succeed no matter which* $S \in \mathcal{C}_2$ *and which* $\epsilon$ *for* $S$ *player* $T$ *choose. Player* $A$ *would have to be announced that he won, otherwise he would not know.*

The examples above are similar in nature to scientific inquiry, Nature chooses a reality $S^T$ from a class $\mathcal{C}$ that is constrained by established theory. The sequence of information that is revealed step-by-step to the scientist in some order represent his observations of the phenomena under study. Success consists in ultimately stabilizing on an hypothesis $S$.

Continuing with the numerical example, more realism comes from limiting members of $\mathcal{C}$ to effectively enumerable subsets of $\mathbb{N}$, named via the programs that enumerate them. Scientists can then be interpreted as computable functions from data to such names. In the same spirit, data-acquisition may

converted to a less passive affair by allowing the scientist to query Nature about particular members of $\mathbb{N}$.

The constraints concerning a learning problem can change from one situation to another and a great variety of paradigms have been analyzed, to mention some:

1. The success criterion can be relaxed or tightened,
2. the data can be partially corrupted in various ways,
3. the computational power of the scientist can be bounded,
4. efficient inquiry can be required,
5. learners can be allowed to work in groups (teams).

Observe that in example 1, our learner $A$ identified $S^T$ in finitely many steps. When a learning function $L$ can identify each $S$ in a class $\mathcal{C}$ of languages in finitely many steps, we say that $L$ *finitely identifies* class $\mathcal{C}$. In example 2, the game never stops, so that the learner can continue guessing infinitely many times. However as we explained before, there is a strategy for player $A$ which can make him win the game in the limit. When a learning function $L$ can identify in the limit each $S$ in a class $\mathcal{C}$ of languages, we say that $\mathcal{C}$ is *identifiable in the limit* by $L$. That said, many models of learning have been developed in formal learning theory, Finite Identifiability, $PAC$, Identifiability in the limit; to mention some.

## 2.3 Identifiability in the limit

In this thesis we will only focus on one very well studied framework in the computational learning field introduced by Gold in 1967: *Identification in the limit*. This model describes a situation in which learning is a never ending process. The learner is given information, builds a hypothesis, receives more information, updates the hypothesis, and so on. The learner can make multiple guesses (even infinite) which guarantees the existence of a reliable strategy that allows for convergence to a correct hypothesis for every element of the class. Example 2 described in the previous section illustrates the idea behind this learning framework.

The exact moment at which a correct hypothesis has been stabilized is not known to the learner and in most cases it is not computable, however there is certainty that at some point the learner will converge to one hypothesis. This setting may seem an unnatural process and completely abstract since it seems that one can study the fact that we are learning a *concept* but not that we have finished learning it. Such learning setting provides some useful insights of the learning problem under consideration. As a matter of fact, learning a language in reality is like this, we also do not know when we are done with it.

Valiant's definition and approach for learnability would also have been *ad hoc* for the problem we are addressing in this thesis, so let us provide our personal motivation for choosing Gold's definition of learnability: First, because we observed a direct analogy we wanted to embrace between Gold's implications to a child, *who on the basis of finite samples learns to creatively use a language, by inferring an appropriate set of rules*; and the learning problem we want to address: *someone who on the basis of finite samples learns creatively use a language of proofs, by inferring an appropriate set of inference rules*. We can also point out similar analogies from Valiant's work; however since the one from Gold's was the one we encountered first, we thought we should be *fair to him* on this respect. Second, because we believe qualitative approaches can provide interesting insights without involving probabilities. In any case, we still believe that very interesting and maybe more powerful results can be obtained for the learning problem under consideration by using Valiant's definition of learnability.

Now we present some formal definitions concerning identification in the limit.
Identification in the limit of a class of languages is defined by the following chain of conditions.

**Definition 5** *(Gold (1967)) A learning function $L$:*

1. *identifies $S_i \in \mathcal{C}$ in the limit on $\epsilon$ iff, for co-finitely many $m$, $L(\epsilon \upharpoonright m) = i$;*

2. *identifies $S_i \in \mathcal{C}$ in the limit iff it identifies $S_i$ in the limit on every $\epsilon$ for $S_i$;*

3. *identifies $\mathcal{C}$ in the limit iff it identifies in the limit every $S_i \in \mathcal{C}$.*

*We will say that a class $\mathcal{C}$ is identifiable in the limit iff there is a learning function $L$ which identifies $\mathcal{C}$ in the limit.*

A characterization theorem provided by (16 - Angluin), says that each set in a class that is identifiable in the limit contains a special finite subset $D$ that distinguishes it from all other languages in the class.

**Definition 6** *(Angluin 1980). A set $D_i$ is a finite tell-tale set for $S_i \in \mathcal{C}$ if;*

1. *$D_i \subseteq S_i$,*

2. *$D_i$ is finite, and*

3. *for any index $j$, if $D_i \subseteq S_j$ then $S_j \nsubseteq S_i$.*

Identifiability in the limit can be then characterized in the following way.

**Theorem 1** *(Angluin 1980). An indexed family of recursive languages $\mathcal{C} = \{S_i | i \in \mathbb{N}\}$ is identifiable in the limit from positive data iff there is an effective procedure $\mathcal{D}$, that on input $i$ enumerates all elements of a finite tell-tale set of $S_i$.*

In other words, each set in a class that is identifiable in the limit contains a finite subset that distinguishes it from all its subsets in the class. For the *effective* identification it is required that there is a recursive procedure that enumerates such finite tell-tale sets.

Some important early results of Identification in the limit can be summarized in the table below extracted directly from Gold's paper (5).

| Information Presentation | Class of languages |
|---|---|
| Anomalous text | Recursively enumerable<br>Recursive |
| Informant | Primitive recursive<br>Context sensitive<br>Context free<br>Regular<br>Superfinite |
| Positive | Finite languages |

Figure 2.1: Gold's results.

As table in Figure 2.1 shows, none of the four language classes in the Chomsky hierarchy is learnable from positive data. In fact, the only class that is learnable from possitive data is completely trivial, since its members are all of finite cardinality [2]. This restricted nature of the stream of data (the availability of positive evidence and the lack of negative evidence) is often referred to as the *poverty of the stimulus*. Gold also considered a model in which the learner is provided with both positive and negative data. In this case, an *oracle* or informant can be consulted by the learning function. This oracle tells whether or not a sentence belongs to the target language. In this case, learning turns out to be much easier.

## 2.4 Formal learning theory and cognition

All the discussion above leads to a simple description of the core of formal learning theory: construction of a diverse collection of approaches to the mathematical modeling of learning. But what does this theory of *learning* contributes to the study of learning in cognitive science? The main contribution comes with

[2] (26 - Horning) proved that so-called probabilistic context-free grammars can be learned from possitive data only. This result removes the sting of the strict unlearnability results of Gold.

stating possible constraints on what is learnable by different types of idealized mechanism. The most famous example is Gold's learning results by means of identification in the limit which provided a *coherent* explanation for language acquisition in humans. Therefore, implementing formal learning results and procedures of this kind in a cognitive model might provide potential useful insights about human learning. By deriving theoretical results of the possibilities a learning system has for success given certain data.

On the one hand, when studying the phenomena of learning in cognitive science disregarding the insights formal learning theory can provide, may lead to misleading and confusing conclusions. Many discussions and computational models for understanding learning in cognitive science are often not accordingly related to theoretical findings. For instance it may be difficult to determine if a particular model can be extended to more complex cases. On the other hand, cognitive science can provide special considerations when building a mathematical model for addressing a real learning problem, since we would like the model to be close to reality in general. When formal learning theory frameworks and problems are rather distant from cognitive scientific questions, it can become just another specialized branch in mathematics or computer science without a concrete application. Trying to bring together technical formalisms in learning theory with more realistic cognitive scenarios and frameworks is not an easy task. A clear example is again the one of Gold's, his results started a vigorous debate in linguistics which is far from over (27). Its deceptive simplicity has led to its being possibly more often misunderstood than correctly interpreted within the linguistics and cognitive science community. However this was one of the firsts that built bridges between cognitive science and learnability theory.

Cognitive science is mostly concerned with the construction of computational models of specific cognitive phenomena (including learning of all kinds, and of course language acquisition) however almost none of these models address how humans learn to reason deductively. This might be because *reasoning* as a normal daily mental activity is not seen as something humans *learn*, but more that something humans *do*. Two of the most prominent and well-known theories of human reasoning are: Rips' *Mental Logic* theory together with his PSYCOP algorithm for deductive reasoning and Johnson-Laird's *Mental models* account. Rips defends formal rules as the basic symbol-manipulating operators of cognitive architecture; suggesting that humans are born with an innate "inference rules"- module which by default should produce valid inferences (as occurs in PSYCOP) (8, 28). (29 - Johnson-Laird) claims that reasoning seems to be based on mental models of the states of affairs described by premises. However none of these views provide a deep account for the process of *learning* deductive reasoning. In one of his many replies to Rips arguing in favor of mental models, (30 - Johnson-Laird) gently poses some questions concerning the *acquisition process* for deductive reasoning:

> *Human reasoning is a mystery. Is it at the core of the mind, or an accidental and peripheral property? Does it depend on a unitary system, or on a set of disparate modules that somehow get along together to enable us to make valid inferences? And how is deductive ability acquired? Is it constructed from mental operations, as Piagetians propose; is it induced from examples, as connectionists claim; or is it innate, as philosophers and "evolutionary psychologists" sometimes argue?*

These theories also lack of an extensive analysis concerning individual differences in reasoning schema leading individuals to produce "erroneous inferences". However they do realize this phenomena and the importance of studying it. Johnson-Laird expresses the following;

> *...erroneous conclusions should be consistent with the premises rather than inconsistent with them, because reasoners will err by basing their conclusions on only some of the models of the premises. They will accordingly draw a conclusion that is possibly true rather than necessarily true. . . .*

while (28 - Rips) argues:

> *As mentioned earlier, errors can arise in many ways, according to the theory, but, for these purposes, let's distinguish errors that stem from people's initial misunderstanding of the premises and those that stem from later parts of the deductive process – for example, priming of conclusions by the premises or misapplication of logical rules...*
>
> *... deduction theories must choose which errors to explain internally and which to explain as the effects of other cognitive processes (e.g., comprehension or response processes). There are*

*certainly sources of systematic error that PSYCOP doesn't explain internally and, likewise, sources that Johnson-Laird's theory can't explain.*

Other theories from the Bayesian school, have models of reasoning that almost by definition include learning (in a specific Bayesian sense) as the key ingredient such as the work of (31 - Goodman) and (32 - Frank and Goodman). For instance, Goodman argues that the validity of a deductive system is justified by its conformity to good deductive practice. The justification of rules of a deductive system depends on our judgments about whether to reject or accept specific deductive inferences. Thus, for Goodman, the problem of induction dissolves into the same problem as justifying a deductive system. Based on this, Goodman claims that Hume was on the right track with habits of mind shaping human reasoning; supporting the view which says that which scientific hypotheses we favour depend on which predicates are "entrenched" in our language (32). In a similar fashion, we could say our results suggest that which inferential systems we favour depend on which *interpretations* of the rules of inference are "entrenched" in our reasoning machinery by our exposure with related information. Later on, (31 - Goodman) argues in favor of Bayesian methods, saying that they have a sound theoretical foundation and an interpretation that allows their use in both inference and decision making when evaluating the chances of a given conclusion to be right or wrong.

There remain fundamental questions about the capabilities of different classes of cognitive theories and models concerning human reasoning and human learning, and about the classes of data from which such models can successfully learn. In this thesis, we will *partially* address some aspects of these questions. Our model will suggest that there are basic parts of this *inferential* human mechanism that are intrinsic; as there are other parts which need to be *learned* by means of how the information is presented and by implementing relevant examples. Every theory of logical reasoning comprises a formal language for making statements about objects and reasoning about properties of these objects. This view of human reasoning is very general (and in some sense restrictive). Logic has deep relations with knowledge structure, semantics and computation. Since deduction is in some sense a human computation, it seems feasible to express our models of learning a *system for reasoning* as an abstract computational procedure at the level of formulas and proofs.

## 2.5 Conclusions

In this chapter we pose the basic notions involved in formal learning theory, presenting the main idea behind it by means of examples that faithfully represent its flavour. We focused on the learning model developed by Gold, identification in the limit, which was originally developed for studying learnability of classes of languages. Finally we discussed some aspects of formal learning theory, its implications for cognitive models for learning; emphasizing the importance of collaboration between these two fields of study.

# Chapter 3

# The many faces of Natural Deduction

## 3.1 Introduction

A logical language can be used in different ways. For instance, a language can be used as a proof system (or deduction system); that is, to construct proofs or refutations. This use of a logical language is called proof theory. In this case, a set of facts called axioms and a set of deduction rules (inference rules) are given, and the object is to determine which facts follow from the axioms and the rules of inference. In this case, one is not concerned with the meaning of the statements that are manipulated, but with the arrangement of these statements, the correct use of the rules; and specifically, whether proofs or refutations can be constructed. In this sense, statements in the language are viewed as cold facts, and the manipulations involved are purely mechanical. In spite of this, having the right interpretation of the usage of the inference rules is a crucial factor for a correct proof. Moreover, finding a proof for a statement requires creativity.

In the first two sections of this chapter we will discuss and analyze the main features of the proof system *Natural Deduction* (ND). In Section 3.3 we address several ways of representing an inference systems, especially for the natural deduction system and the possible implications each representation may carry. We will end up concluding that an hybrid of the three forms presented is the representation we are aiming for, in order to best characterize the alternative inference systems concerned for the learning space.

What is natural deduction for? Natural deduction is used to prove that some argument is correct. For example: If I say: "In the winter it's cold, and now it is winter, so now it's cold". A listener would start thinking and processing what I just said to finally reply: "OK, it follows". In simple words, given a supposition "if all this happens, then all that also happens as well", natural deduction allows us to say "yes, that's right". But why is such mathematical proof mechanism needed for simple real-life situations? Well it is not always so easy to check validity of a reasoning. Take the following example:

*"If you fail a subject, you must repeat it. And if you don't study it, you'll fail it. Now suppose that you aren't repeating it. Then, either you study it, or you are failing it, or both."*

This reasoning is valid and it can be proven with natural deduction. Note that you do not have to believe nor understand what you are told. Why is that possible? For example, if I say: "Blablis are shiny and funny; a pea is not shiny, so it isn't a blablis". Even if you don't know what am I talking about, you must be sure that the reasoning seems correct. Therefore natural deduction as a verification mechanism for valid inferences given certain premises, disregards the meanings or interpretations of the words and phrases and just pays attention to the connectives, order, and structure of these words and phrases in the reasoning procedure. Verification mechanisms of this kind are clearly very useful in logic and mathematics; but also in real life complex reasoning tasks.

As trivial as it might sound, it is worth mentioning that natural deduction cannot prove invalid statements (there are some methods for doing so). Natural deduction cannot succeed on proving expressions like "If it is Sunday it is not Monday; today it is Sunday so it is also Monday".

### 3.1.1 History

A historical motivation for the development of system of natural deduction for propositional logic was to define the meaning of each connective syntactically, by specifying how it is introduced and eliminated from a proof. There is a wide variety of interesting and in many ways useful approaches to logic specification, neither of them comes particularly close to capturing these practice of mathematical proofs. This was Gentzen's point of departure for the design of a formal system capturing a more realistic process of mathematical reasoning (33, 34). Natural deduction rules of inference would fix interpretations of the connectives by specifying their functional roles in a proof. According to (35 - Jaśkowski)[1], Jan Łukasiewicz has raised the issue in his 1926 seminars that mathematicians do not construct their proofs by means of an axiomatic theory (the systems of logic that had been developed at the time, as in the Hilbert tradition) but rather made use of reasoning methods; especially they allow themselves to make "arbitrary open assumptions" and see where they lead.

With reference to Gentzen's work, (34 - Prawitz) made the following remarks on the significance of natural deduction.

> *... the essential logical content of intuitive logical operations that can be formulated in the languages considered can be understood as composed of the atomic inferences isolated by Gentzen. In this sense that we may understand the terminology* natural *deduction.*
>
> *Nevertheless, Gentzen's systems are also natural in the more superficial sense of corresponding rather well to informal practices; in other words, the structure of informal proofs are often preserved rather well when formalised within the systems of natural deduction.*

The idea that the meaning of connectives can be defined by inferential role has been wide spread and dominant amongst the logic and mathematical community. It is important in proof-theoretic semantics for intuitionistic logic which Gentzen (unlike Jaśkowski) considered besides from the one for classical logic. It also resides prominently in the discussion of the characterization of "the proper form of rules of logic" in terms of introduction and elimination rules for each of the logical connectives as the key for describing not only what was *meant* by a logical connective, but also what a *true system in logic* should look like (36).

Later on, in the 1970's, when theories of reasoning started to be popular among psychologists, a number of theorists adapted natural deduction rules to explain human deductive reasoning (37, 38, 8, 39). In these accounts, humans are thought to apply formal rules to mental representations of propositions so as to reach desired or interesting conclusions. This view also fits well with Fodor's claim that there is a language of thought (40, 41). Fodor argues that cognitive performance requires an internal system of language-like representations and formal syntactic operations which can be applied to these representations. This strong claim suggests that language provides the metaphor by which theorists can understand and model cognition. Thus, if the cognitive system has an overall language-like architecture then it makes sense to model deductive reasoning by specifying mental rules (comparable to the inference rules of natural deduction) that work upon language-like mental representations.

Nowadays, the most used characterizations for natural deduction system are: the tree representation by (33 - Gentzen), and the linear representation developed originally by (35 - Jaśkowski) and refined later by Fitch (42). Another recent one (not very common) is with formulas-as-types and proofs-as-programs, as in simply typed $\lambda$ calculus. As a matter of fact due to the Curry-Howard isomorphism theorem, we know that natural deduction system and simply typed $\lambda$ calculus are two different names for the same system (43).

### 3.1.2 How does it work?

So how does natural deduction work? When we are asked to prove the validity of $\Gamma \vdash A$, where $\Gamma$ is a group of formulas separated by commas called *premises*, and $A$ is a single formula. We start assuming that all formulas in $\Gamma$ hold, and, by continuous application of nine proof rules, we can go on discovering which other things hold. Our goal is to discover that $A$ holds; so once we achieve that, we can stop working. This is something very important to consider, since we could always continue applying the rules obtaining an infinite amount of valid inferences; but this is not a realistic scenario. The number of

[1] In his 1934's paper, Jaśkowski argues that he developed independently a system equivalent to the one of Gentzen's natural deduction and that he presented it to the First Polish Mathematical Congress in 1927.

inferences we are obtaining is virtually bounded by the aim of reaching the desire conclusion. So, if one is not following the right way towards the target conclusion, one might miss it.

Sometimes our set of premises will be empty. Hence, we will have to make suppositions: "well, I'm not sure that $A$ holds, but if it holds that $C$, then without a doubt $A$ is the case". This simple example illustrates how by making suppositions we can obtain that statements like *when assuming $C$ it follows that $A$* hold.

Natural deduction is a collection of formal systems that use a common structure for their inference rules. The specific inference rules of a member of such a family characterize the theory of a logic. Usually a given proof calculus encompasses more than a single particular formal system, since many proof calculi are under-determined and can be used for radically different logics. For instance Natural deduction serves as a proof system for classical logic (CL), however with few modifications it can serve as a proof system for intuitionistic logic (IPC).

## 3.2 Natural deduction proof system for propositional logic

### 3.2.1 Basic definitions

Imagine someone says: "It is raining", a moment later the speaker continues "If it is raining then the sidewalk is wet". After a moment he concludes "It is raining and the sidewalk is wet". We can use symbols to represent what the speaker just said: $P$:= *It is raining*; $Q$:= *The sidewalk is wet*; $P \to Q$:= *If it is raining then the sidewalk is wet*; and $P \wedge Q$:= *It is raining and the sidewalk is wet*. Note that $\to$ and $\wedge$ represent the connectives $IF \ldots THEN$ and $AND$, respectively.

In accordance with the order of utterances, the reasoning went as follows:

1. $P$
2. $P \to Q$
3. $P \wedge Q$

It seems that there is something implicit when going from step 2 to step 3. In the reasoning process of making inferences, a finite list of steps is specified. Each step in the reasoning process is constructed by applying certain rules concerning the way in which these steps can be put together in order to build derivations. Using our example above, the complete reasoning process is as follows:

1. $P$ premise,
2. $P \to Q$ premise,
3. $Q$ because we have $P$ and from $P$ we can obtain $Q$,
4. $P \wedge Q$ since we have both $P$ and $Q$.

Clearly there was a *rule* applied in step 3 in order to obtain $Q$ and another rule applied in step 4 to obtain $P \wedge Q$ in this reasoning process. But which rules?; and, how can we know when to apply them?

The following questions arise naturally: a) *When can we infer as a conclusion, a formula which main connective is $\wedge$ (as in step 4)?* and b) *What can we infer from formulas which main connective is $\to$ (as in step 3)?* . In propositional logic we want to provide answers to those kinds of questions for every logical connective; and natural deduction seems to provide direct answers.

The reasoning steps that correspond to the answer of question a) for each connective are indicated in the *introduction rule*. The answer for question b) is indicated in the *elimination rule*.

Certain forms of judgments frequently recur and have therefore been investigated in their own right, prior to logical considerations. We will use hypothetical judgments of the form: *"$C$ under hypothesis $B$"*. We consider this judgment evident if we are prepared to make the judgment $C$ once provided with evidence for $B$. Formal evidence for a hypothetical judgment is a hypothetical derivation where we can freely and openly use the assumption $B$ in the derivation of $C$. We will often refer to hypotheses like $B$ as *open assumptions*. Note that hypotheses of this kind need not be used, and could be used more than once.

Formal evidence for a judgment in form of a derivation is usually written in two-dimensional notation:

$$\begin{array}{c}\mathcal{D}\\ J\end{array}$$

where $\mathcal{D}$ is a formal derivation.

A hypothetical judgement is written as,

$$\begin{array}{c}J_1^u\\ J_2\\ .\\ .\\ .\\ J_n\end{array}$$

where $u$ is a label which identifies the hypothesis $J_1$ as an open assumption. Labels are often used to guarantee that open assumptions which are introduced during the reasoning process are not used outside their scope.

Consider $\mathcal{L}$ the same language as for propositional classical logic composed by propositional letters $p, q, etc.$, constants $\bot, \top$ representing *truth* and *falsum;* logical connectives $\wedge, \vee, \rightarrow$; and one argument operator $\neg$ representing the natural relation between one expression and another $AND$, $OR$, $IF \ldots THEN$ and $NO$ respectively.

**Definition 7** *The language of propositions is built up from propositional letters as*

$$\textit{Propositional formulas } A ::= p \,|\, A_1 \wedge A_2 \,|\, A_1 \rightarrow A_2 \,|\, A_1 \vee A_2 \,|\, \neg A \,|\, \bot \,|\, \top$$

*We will use $FORM$ to denote the set of propositional formulas.*

The semantics of each symbol we have:

- For $\wedge$ we read $AND$ we have: $A \wedge B$ holds if and only if $A$ holds and $B$ holds.
- For $\vee$ we read $OR$ we have: $A \vee B$ holds if and only if either $A$ holds, $B$ holds, or both hold.
- For $\rightarrow$ we read $IF \ldots THEN$ we have: $A \rightarrow B$ holds if and only if whenever $A$ holds, so does $B$.

We still consider the usual order-priority of connectives $\rightarrow_1, \vee_2, \wedge_2, \neg_3$. Observe that $\wedge$ and $\vee$ have the same priority which is higher than $\neg$. When you see an expression, you must be able to recognize if it is an implication, a disjunction, a conjunction, or a negation. For instance, $A \wedge B \rightarrow C$ is an implication not a conjunction, because $\rightarrow$ has priority over $\wedge$.

Certain structural properties of proofs are tacitly assumed, independently of any logical inferences. In essence, hypothetical judgments work as follows: 1) If we have a hypothesis $A$ then we can conclude $A$, 2) hypotheses need not be used, 3) hypotheses can be used more than once. We will assume that from all inference systems discussed in this thesis at least natural deduction obeys the monotonicity rule:

**Definition 8** *Let $A, B \in FORM$ and $\Gamma$ a multiset of elements in FORM. The* monotonicity structural rule *is:*

- *(Monotonicity)* [2] $\dfrac{\Gamma \vdash B}{\Gamma, A \vdash B}$

### 3.2.2 Elimination and introduction rules

The inference rules that introduce a logical connective in the conclusion are known as introduction rules. These rules express what kind of inferences are valid with a logical connective given certain premises. The elimination rule for the logical connective tells what other subformulas we can deduce from a complex

[2] Monotonicity in human reasoning has been questioned by several researchers. (44 - Pfeifer and Kleiter) argue the following: "Monotonicity is a meta-property of classical logic. It states that adding premises to a valid argument can only increase the set of conclusions. Monotonicity does not allow to retract conclusions in the light of new evidence. In everyday life, however, we often retract conclusions when we face new evidence. Moreover, experiments on the suppression of conditional inferences show that human subjects withdraw conclusions when new evidence is presented. Thus, the monotonicity principle is psychologically implausible."

formula. Thus we can say that these rules provide specific insights for the correct interpretation of the logical connectives, also seen as the human reasoning connectives.

Recall that each connective is defined only in terms of inference rules without reference to other connectives. This feature of independence between the connectives, means that we can understand a logical system as a whole by understanding each connective separately. It also allows us to consider fragments and extensions of propositional logic directly.

The introduction and elimination rules for each connective are the following:

**Implication:** To derive that $A \to B$ holds we assume $A$ holds as a hypothetical judgment and then derive that $B$ also holds. So we obtain the following introduction rule denoted by $\to I$:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

The elimination rule expresses that whenever we have a derivation of $A \to B$ and also a derivation of $A$, then we can also have a derivation of $B$. We have the following elimination rule for implication denoted $\to E$:

$$\frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

**Conjunction:** $A \wedge B$ should hold if both $A$ and $B$ hold. Thus we have the following introduction rule denoted $\wedge I$:

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

Now, to recover both $A$ and $B$ if we know that $A \wedge B$ holds, we need two elimination rules denoted $\wedge E^r$ and $\wedge E^l$ respectively:

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \qquad \Bigg| \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$$

**Disjunction:** The introduction rule denoted by $\vee I^r$ says that whenever we have a derivation of $A$, the same derivation is enough for having that $A \vee B$ holds. Similarly for $B$ denoted by $\vee I^l$:

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \Bigg| \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$$

The elimination rule for disjunction denoted by $\vee E$ is not as simple as the rest since having that $A \vee B$ holds, does not provide any insights about $A$ or $B$ separately. The way to proceed is with a derivation by cases: we prove a possible conclusion $C$ under the open assumption $A$ and also show $C$ under the open assumption $B$. We then conclude $C$; since when either $A$ or $B$ are open assumptions $C$ follows. Note that the rule employs two hypothetical judgments, one with open assumption $A$ and another one with open assumption $B$.

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}$$

**Negation:** The introduction rule for negation denoted by $\neg I$ expresses that if when assuming that $A$ holds we always obtain a proof for any propositional formula $D$, then we will be able to derive a contradiction. Thus the negation of $A$ should hold.

$$\frac{\Gamma, A \vdash D}{\Gamma \vdash \neg A}$$

For the elimination rule, denoted by $\neg E$, an analogous argument is conceived: if we know that $\neg A$ holds and $A$ holds then we can conclude that any formula $D$ also holds.

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash \neg A}{\Gamma \vdash D}$$

**Truth:** There is only an introduction rule for $\top$ denoted by $\top I$:

$$\frac{}{\Gamma \vdash \top}$$

Since we put no information into the proof of $\top$, we know nothing new if we have an assumption, therefore we have no elimination rule.

**Falsehood:** We should not be able to derive falsehood, so there is no introduction rule for $\bot$. Thus, if we can derive falsehood, we can derive everything. We have the elimination rule for falsum denoted as $\bot E$:

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash D}$$

When doing a formal proof, the introduction and elimination rules are not to allow the learner to write anything he wants, but to help him use a premise or an open assumption to create a dependable conclusion with a concrete operator. That is why, if you have $P$, you can't say "now I do negation introduction and get $\neg P$, which is what I needed". There are some requisites for each rule, and if you do not fulfill them, you cannot apply that rule. The proper way of using the rules can be difficult to grasp at the very beginning, and an effective way of *proving* requires creativity, patience and only gets learned with regular practice.

Something that is worth mentioning is that the separation of the notion of judgment and proposition and the corresponding separation of the notion of evidence and proof sheds new light on various styles that have been used to define logical systems. The main judgment of natural deduction is "$C$ holds" written as $C$ holds, from hypotheses "$A_1$ holds", ..., "$An$ holds". In contrast, an axiomatization in the style of Hilbert for example, arises when one defines a judgment "A is true" without the use of hypothetical judgments (45, 3). Such a definition is highly economical in its use of judgments, which has to be compensated by a liberal use of implication in the axioms. There are many presentations which are highly economical and do not need to seek recourse in complex judgment forms (at least for the propositional fragment). However proofs not only in mathematics but in real life often require many hypotheses.

### 3.2.3 Local reduction and local expansion

Introduction and elimination rules are not independent on each other, an introduction and elimination rule for each connective must match in a certain way to guarantee that the rules are meaningful and the overall system can be seen as capturing deductive reasoning. A set of formulas is said to be sound if we cannot derive falsehood (from no assumptions) and is complete if every valid formula is provable using the inference rules of the logic. These are statements about the logic as a whole, and are usually tied to some notion of a model. However, there are local notions of consistency and completeness that are purely syntactic checks on the inference rules, and require no appeals to models.

The first is a local soundness property expressing that if we introduce a connective and then immediately eliminate it (with the corresponding rules), we should be able to erase this loop in the derivation and find a more direct derivation of the conclusion without using the connective. If this property fails, the elimination rules are too strong since they allow us to conclude more than we should be able to know.

The second is a local completeness property expressing that we can eliminate a connective in a way which retains sufficient information to recover it by an introduction rule. If this property fails, the elimination rules are too weak since they do not allow us to conclude everything we should be able to know. We provide evidence for local soundness and completeness of the rules by means of local reduction and expansion judgments, which relate proofs of the same propositional formula.[3]

[3]For more information about *local expansion and local reduction* and some examples of proof detours we recommend to the reader to look into (34, 3, 46, 47 - or any other proof theory text book).

### 3.2.4 Formal proofs and daily life deductive reasoning

Formal proofs are interesting and in many ways useful approaches to logic specification, however some proof characterizations do not comes particularly close to capturing the natural practice of either mathematical or daily life reasoning. But why is it the case with natural deduction? Because the procedures to be applied are very similar to the ones people use while reasoning (8, 37, 38). You can see *that* in most solved exercises in every proof theory manuals. Express the sequents by words, tell them to someone, and after some time it is often the case that he/she will be saying "of course it's like that, since ...". You will see that anyone is able to understand (at some extent) how to use the nine derivation rules, even without knowing their name or existence. Forget about introduction and elimination rules and think normally, changing the letters to simple expressions if necessary (Brunett and Medin revise and discuss in their paper *Reasoning across cultures*, several empirical studies concerning the idea of "universality" in logical reasoning, (48)). For this reason natural deduction rules, as formal as they might be, seem to mimic (to some extent) quite precisely a wide variety of human reasoning processes.

Apart from the elimination and introduction rules mentioned before, there is more in what respects to the act of *proving* that something holds. A variety of mechanisms can be implemented (and usually they are necessary) as useful tools while proving something. The most important ones are:

- Iteration of hypotheses/premises.
- Introducing open hypothetical assumptions.
- Sub-derivations to use for a bigger derivation.
- Reasoning by cases.
- Reduction to absurdum.
- Assuming the contrary of what you are proving.

These proving tools express some ways of human thinking in a wider sense, independently of natural deduction or any other inferential system. Complicated derivations as the ones we regularly encounter in mathematics, logic or in philosophy are not very common, so it seems that we do not need to use most of these tools in quite simple daily-life reasoning processes. However we could say that, to some extent, humans are equipped with such generic reasoning mechanisms; but we restrict ourselves to say that in a very intrinsic way. Further on, we will see that our model suggests that humans can acquire misleading inference systems which lead people to "invalid" inferences according to the normative way of reasoning (according to classical logic).

As we already mentioned, there are different forms to represent a proof in natural deduction. The two most common proof representations are: the tree representation and the linear representation. In the latter, the order of the derivation process is relevant. It is easy to find different ways for writing a derivation in the literature and even many proof techniques. It seems that often logicians pick the most suitable one depending on the current mathematical/computational goal. These various ways of characterizing proofs also bring different cognitive considerations one should take into account when studying the learning process. Some studies suggest that the order of how information is provided play a significant role while reasoning, supporting the selection of the linear representation as the most cognitively adequate (8).

## 3.3 Exploring different representations for an inference system

The learning space we will focus in our learning problem should represent possible misinterpretations of the deductive rules from natural deduction. We want the learning space to be a set containing the possible alternatives people can use while reasoning, but in principle such space is arbitrary and can be hard to define.

An important issue for our study is to decide how to represent the inference systems in the learning space. While doing so, some questions arise naturally: *Which representation fits best our intuitions for an inference system? What kinds of cognitive implications will carry?* The whole nature of the model itself may be affected by this choice. The way we represent our learning objects is a crucial factor for our model since choosing one representation over another one involves some changes on the framework

considerations and results.

In this section we will discuss several ways of representing the inference systems which are sets of inference rules; particularly the natural deduction system addressing the implications each representation may carry. We will discuss three possible ways of characterizing natural deduction: 1) as a *grammar*, where the *language* will be the set of complete proofs that is generated by propositional formulas; 2) as similar to an *axiomatic system*; and the usual form 3) as *scheme of rules* operating as rules for reasoning. We will evaluate the advantages and disadvantages in each case; and conclude that an hybrid of these three cases (in order to preserve as much advantages as possible) is what we are aiming for the best characterization.

Further on in this work we will discuss in detail the mathematical nature and features of these alternative systems. But for now we will just discuss and evaluate the desired features we want them to have.

**Definition 9** *We will make use of the following notation:*

- $\mathcal{R}$ *denotes the class we will focus on in this learning problem, which is the class of inference systems that are misinterpretations of the natural deduction inference set. We will often call* $\mathcal{R}$ *the learning space or the set of hypotheses.* $\mathcal{R}$ *will also contain the inference set corresponding to natural deduction denoted by* $R_{ND}$*.*
- $R \in \mathcal{R}$ *denotes an inference system in class* $\mathcal{R}$*.*
- $R^T$ *will denote the target set, i.e, the inference system that needs to be learned. We will often refer to* $R^T$ *the target proof system.*

### 3.3.1 Natural deduction as a grammar

A formal grammar is a set of rules for rewriting strings plus an indicator symbol from which rewriting starts. A grammar is usually thought of as a language generator. However, it can also sometimes be used as the basis for a "recognizer", i.e. a function that determines whether a given string belongs to the language or if it is grammatically incorrect. To describe such recognizers, formal language theory uses separate computational mechanisms, known as automata (for more information about formal languages, automata theory and computational complexity we invite the reader to look into (49 - Hopcroft and Ullman)).

A grammar mainly consists of a set of rules for transforming strings. To generate a string in the language, one begins with a string consisting of only a single start symbol. The production rules are then applied in any order, until we obtain a string which contains neither the start symbol nor designated non-terminal symbols. A production rule is applied to a string by replacing one occurrence of the production rule's left-hand side in the string by that production rule's right-hand side, i.e. one step at a time. The language formed by the grammar consists of all distinct strings that can be generated in this manner. Any particular sequence of production rules on the start symbol generates a distinct string in the language. If there are essentially different ways of generating the same single string, the grammar is said to be ambiguous.

Formally, following the classical definition of generative grammars first proposed by Noam Chomsky, a grammar is defined as follows.

**Definition 10** *A grammar* $G$ *is the tuple* $(\Delta, \Sigma, P, S)$ *such that:*

- $\Delta$ *is a finite set of nonterminal symbols, that is disjoint with the strings formed from* $G$*.*
- $\Sigma$ *is a finite set of terminal symbols that is disjoint from* $\Delta$*.*
- $P$ *is a finite set of production rules, such that each rule is of the form*

$$(\Sigma \cup \Delta)^* \Delta (\Sigma \cup \Delta)^* \to (\Sigma \cup \Delta)^*$$

*where* $*$ *is the Kleene star operator and* $\cup$ *denotes the usual set union.*

- *A distinguished symbol $S \in \Delta$ that is the start symbol, also called the sentence symbol.*

Each production rule maps one string of symbols to another, where the first string (the "head") contains an arbitrary number of symbols provided at least one of them is a nonterminal. In this process, the productions are used as rewriting rules.

**Definition 11** *Given a grammar $G = (\Delta, \Sigma, P, S)$, the (one-step) derivation relation $\Rightarrow_G$ associated with $G$ is the binary relation $\Rightarrow_G \subseteq \Delta^* \times \Delta^*$ defined as follows: for all $\alpha, \beta \in \Delta^*$, we have $\alpha \Rightarrow_G \beta$ iff there exist $\lambda, p \in \Delta^*$, and some production $(A \to \gamma) \in P$, such that*

$$\alpha = \lambda A p \text{ and } \beta = \lambda \gamma p.$$

We will call this $\Rightarrow_G$ derivations g-derivations.

**Definition 12** *Given a grammar $G = (\Delta, \Sigma, P, S)$, the binary relation $\Rightarrow_G^*$ denotes the reflexive transitive closure of the binary relation $\Rightarrow_G$.*

**Definition 13** *A string $\alpha \in \Delta^*$ such that $S \Rightarrow^* \alpha$ is called a sentential form, and a string $w \in \Sigma^*$ such that $S \Rightarrow^* w$ is called a sentence. A g-derivation $\alpha \Rightarrow^* \beta$ involving $n$ steps is denoted as $\alpha \Rightarrow^n \beta$.*

**Definition 14** *Let $G$ be a grammar. The set $\{w \in \Sigma^* \mid S \Rightarrow_G{}^* w\}$ of all sentences we obtained by a finite number of steps from the start symbol $S$ is the* language of the grammar *which will be denoted as $L(G)$.*

To illustrate the definitions above consider the following examples:

**Example 3** *Let $G$ be such that $\Delta = \{S\}$, $\Sigma = \{a, b\}$ and the following production rules in $P$:*

1. $S \to aSb$
2. $S \to ba$

*we start with $S$ and we can choose a rule to apply to it. Lets choose rule 1 to apply first, we obtain the string [aSb]. If we then choose rule 1 again, we replace $S$ with [aSb] and obtain the string [aaSbb]. Now if we change to rule 2, we replace $S$ with [ba] and obtain the string [aababb], and we stop the process. The language we obtain with this grammar is $\{a^n bab^n : n \geq 0\}$.*

**Example 4** *$G = (S, +, *, (, ), a, \{+, *, (, ), a\}, P, S)$, where $P$ is the set of rules*

- $S \to S + S$
- $S \to S * S$
- $S \to (S)$
- $S \to a$

*The language we obtain with this grammar is the set of all arithmetic expressions.*

Suppose we chose our sets of rules $R \in \mathcal{R}$ to be defined as grammars. What we want to be the language of $R$ in our model are proofs or sets of arguments. If we want the language to be sets of arguments, the rules in $R$ will be approximate translations of the possible inference rules in natural deduction. So inference rules seen as grammatical rules. The g-derivations will play the role of derivations using a set of rules, i.e. g-derivations will play the role of the deducting steps. If we chose our languages to be proofs, the rules in $R$ should express restrictions in order to obtain a valid proof. In this setting the g-derivations will play the role of some sort of proofs.

Several questions arise. Can the rules for natural deduction be expressed as production rules for a grammar? Can natural deduction can be seen as a formal grammar? What is its complexity, e.g, is it context-free?

Further on in this thesis we will see that, to some extent, natural deduction rules behave as rules in a formal grammar taking the form of inference rules that behave almost as functions.

Some remarks, questions or things to have under consideration:

- Can we build a context-free grammar for natural deduction?

- Ambiguity: A context-free grammar is called *ambiguous* if there exists a string that can be generated by two different left-most derivations. Note that in ND proof system we can obtain the same inference with different rule applications. Therefore in our ND grammar we want to mimic this feature.
- Language equality: Given two CFGs, do they generate the same language? The undecidability of this problem is a direct consequence of the previous: it is impossible to even decide whether a CFG is equivalent to the trivial CFG defining the language of all strings. Note that this is relevant for our model since the teacher needs to compare the resulting set of inferences (learner's utterances) or 'language' of the learner with the one of his own which is the correct set of inferences for the given premises. Furthermore to compare two proof systems, the one of the learner and the target proof system.
- Language inclusion: Given two CFGs, can the first one generate all strings that the second one can generate? If we translate this problem for two inference systems, can the first inference system generate all proofs the second one can generate?

To sum up: One may question whether formal grammar is a good representation for natural deduction at all. Grammars might have little to say about the real complex insights of a proof. But that is not to say that *grammaticality* and truth conditions are impossible to combine. Furthermore, that a grammar would take the form of a logic and processing would take the form of deduction and vice versa. By doing so, we make use of the perspective of language engineering and the scientific perspective of logic in order to guide our intuition making it possible. Automated language processing divides mainly into parsing (computing meanings/signifiers from forms/signifiers) and generation (computing forms/signifiers from meanings/signifiers). When seeing logic as a grammar, these computational tasks take the form of deduction-as-parsing and making inferences as generation. A grammatical derivation might not be able to capture some relevant pieces of the proving process itself that a natural deduction derivation contains. For instance how can we represent a grammatical derivation when an open hypothesis has been introduced? such hypotheses will be eliminated via an appropriate rule. Maybe it is not necessary to address exactly how the process will be processed by the learner, since the set of rules are precisely "derivation rules" in order to obtain possible conclusions from a given set of premises. It seems that a g-derivation will only capture the possible rules used in a derivation (these rules can be incorrect or correct).

### 3.3.2 Natural deduction as a set of axioms

An alternative way of representing the elements of $\mathcal{R}$ is by translating them into axioms of propositional logic. The idea of having classical logic axioms as a basis for human rationality was for a very long time the most widely accepted theory involving human reasoning. This view has been controversially discussed amongst others by psychologists and cognitive scientists for the last 40 years. Specially after some relevant experimental studies suggested that humans do not reason following classical logic axioms (Wason selection task, (see 50, 51); Conjunction fallacy, (see 52)).

Nowadays multiple theories of human deductive reasoning are constantly under debate. (29 - Johnson-Laird) has been leading the Mental Models school claiming that people use mental model representations instead of classical logic axioms to produce inferences. Lance J. Rips which is one of the mental logical axioms supporters, in his public statement "Goals for a Theory of Deduction: Reply to Johnson-Laird" , explains his motivation and intentions for setting out a theory of human deductive reasoning that has an approximate scope of first-order logic. (28 - Rips) describes how inferences depend on both sentence connectives and quantified variables. Rips implemented his theory as a computer program called PSYCOP that allowed simulating his theory's claims about the mental process humans follow when making inferences.

The first idea that comes to the mind of a logician when searching for a suitable and simple formal representation of the rules that belong to natural deduction is to represent them as axioms. The advantage of this representation is that natural deduction proof system gets reduced into the conjunction of a finite set of formulas which makes it easier to manipulate. Moreover, this allows us to easily compare two proof systems represented in an axiomatic way.

We propose the following translation for the rules of natural deduction into propositional logic formulas. The square brackets differentiate the premises from the conclusion.

**Definition 15** *$\mathcal{I}$ will denote the set of all the translations defined below for the natural deduction introduction rules.*

*Introduction rules*:

- $(\wedge - Introduction) := [A] \wedge [B] \rightarrow (A \wedge B)$
- $(\vee - Introduction) := [A] \rightarrow (A \vee B)$ and $[B] \rightarrow (A \vee B)$
- $(\rightarrow - Introduction) := [A \Rightarrow B] \rightarrow (A \rightarrow B)$
- $(\neg - Introduction) := [A \Rightarrow \bot] \rightarrow \neg A$
- $(\top - Introduction) := \top$

$\mathcal{E}$ will denote the set of all the translations defined below for the natural deduction elimination rules.

*Elimination rules*:

- $(\wedge - Elimination) := [A \wedge B] \rightarrow A$ and $[A \wedge B] \rightarrow B$
- $(\vee - Elimination) := [(A \vee B) \wedge (A \Rightarrow C) \wedge (B \Rightarrow C)] \rightarrow C$
- $(\rightarrow - Elimination) := [A \wedge (A \rightarrow B)] \rightarrow B$
- $(\neg - Elimination) := [\neg A \wedge A] \rightarrow C$
- $(\bot - Elimination) := [\bot] \rightarrow C$

**Definition 16** *Let $\hat{ND} = \bigwedge \mathcal{I} \wedge \bigwedge \mathcal{E}$. Thus $\hat{ND}$ represents the natural deduction proof system in terms of a conjunction of propositional logic formulas.*

An $\hat{ND}$ derivation will be represented as a classical logic derivation using a fixed set of axioms. The main problem with this setting is that each connective is defined in terms of $\rightarrow$, taking away the modularity feature in defining each connective independently from the rest. Another problem to consider is that the translations for both $A \rightarrow B$ and $A \Rightarrow B$ get reduced to $A \rightarrow B$ only. This is counter-intuitive in many ways. First because $\rightarrow$ and $\Rightarrow$ denote similar but still different "cause-effect" relations between $A$ and $B$. The former corresponds to material implication in logic which is a binary connective that can be used to create new formulas; and concerns the specific truth conditions of such connective. The latter corresponds to a *meta*-connection between two formulas; expressing that any proof for $A$ serves also as a proof for $B$. Thus, material implication ($\rightarrow$) is a symbol at the object level, while logical implication ($\Rightarrow$) is a relation at the meta level. Second because it's not realistic, this extremely abstract representation of hypothesis (premises) and conclusions plus the inference process itself seem too far away from the natural processes humans follow.

### 3.3.3 Natural deduction as a set of scheme-rules

Another representation for $R \in \mathcal{R}$ we want to address is the usual, as classical logic sets of inference rules. Such inference rules will be of the form $\Gamma \vdash C$ where $\Gamma$ is a set of hypothesis and $C$ is a conclusion.

The translation for each natural deduction rule into this form is very straight forward.

**Definition 17** *Let $ND^{\vdash}$ be the set of all rules defined below:*

*Introduction rules*:

- $(\wedge - Introduction) := \{A, B\} \vdash A \wedge B$
- $(\vee - Introduction) := \{A\} \vdash A \vee B$ and $\{B\} \vdash A \vee B$
- $(\rightarrow - Introduction) := \{(A \Rightarrow B)\} \vdash (A \rightarrow B)$
- $(\neg - Introduction) := \{(A \Rightarrow \bot)\} \vdash \neg A$

- $(\top - Introduction) := \vdash \top$

*Elimination rules*:

- $(\wedge - Elimination) := \{A \wedge B\} \vdash A$ and $\{A \wedge B\} \vdash B$
- $(\vee - Elimination) := \{A \vee B, (A \Rightarrow C), (B \Rightarrow C)] \vdash C$
- $(\rightarrow - Elimination) := \{A, (A \rightarrow B)\} \vdash B$
- $(\neg - Elimination) := \{\neg A, A\} \vdash C$
- $(\bot - Elimination) := \{\bot\} \vdash C$

A $ND^{\vdash}$ derivation will be represented as a classical logic derivation using a fixed set of inference rules.

This is the usual way of representing the rules in natural deduction (they are usually written as trees). Simply because captures precisely the desired modularity in defining each connective. However it is not straightforward to "see" how can they be applied in order to build a proof following a reasoning procedure.

### 3.3.4 Conclusion

Natural deduction system is based on the simple judgment "A holds", but relies critically on hypothetical judgments (with open assumptions) mimicking how humans use hypotheses and bound information while making inferences. In addition, it is extremely elegant since it has the great advantage that one can define all logical connectives without reference to any other connective. This modularity feature of connectives agrees with the –to some extent– accepted view that it is not *natural* in mathematical proofs (and for real life reasoning) to interpret one logical connective in terms of other connectives and to not use open assumptions.

We discussed three different ways of representing the rules in natural deduction. Each one of them brings different advantages and disadvantages. We will like to obtain as many advantages from each of these representations as possible, thus we introduce an hybrid version for the natural deduction system we will be addressing in our learning problem. From this one, the rest of the inference systems will be defined in a way that can be thought also as hybrids of these three forms with the aim of trying to capture the most important features each one can provide. Think of the inference systems $R \in \mathcal{R}$ as an interesting dish composed by many different ingredients that provide the necessary flavors for its perfect taste.

# Chapter 4

# The learning space

## 4.1 Introduction

The main goal of this thesis is to propose a learner who: (1) is able to effectively learn a deductive system corresponding to Natural Deduction (2) within the learning process, the learner is expected to disambiguate, i.e., choose one deductive system over other possibilities. The latter property requires that the model includes a class of possible different deductive systems.

The importance of such other, normatively speaking incorrect deductive systems comes from empirical research. The errors made by human subjects in logical reasoning tasks often display patterns, mistakes tend to be systematic. Often subjects seem to posses a faulty reasoning system. Such error patterns have been observed in subjects' performance in Deductive Mastermind implemented within the Math Garden massive online education system by (1 - Gierasimczuk et al.). In their work on conditionals, (2 - Fugard et al.) observed that participants often choose and kept their arbitrary interpretations. In both cases the authors seem to agree on the importance of distinguishing errors that stem from subjects' initial misunderstanding of the premises from those that steam from later parts of the deductive process (for instance, the misapplication of logical rules). This is why we want to address the phenomenon of learning possible different ways of interpreting the logical connectives and its effect on the inferential process. As (53 - Pfeifer and Kleiter) expressed in "The Conditional in Mental Probability Logic":

*For the explanation of typical reasoning, good and bad inferences require a theory of how representations are formed and manipulated.*

Taking this into account, we provide a formal mathematical characterization of the inference systems that represent some possible misinterpretations of the correct inference rules. Such characterization will help us address not only how these rule misinterpretations can occur in a classroom or a conversational environment, but also how they can lead indeed to acquiring different inference systems for reasoning.

The structure of this chapter is as follows: In Section 4.2 we will define natural deduction in new terms: where each rule will be given a class of functions. We will also provide a corresponding proof; and we will show correspondence with the usual system and the version introduced here. In Section 4.3 we will define the possible misinterpretations of natural deduction rules, which constitute a class of possibilities the learner can choose for inference systems from the learning space.

## 4.2 The inference system $R_{ND}$

In this section we will see that natural deduction rules can be formalized in terms of classes of its instances. In this we follow (36 - Garson) definition of an inference system, defining inference rules as classes of its instances; in which rule instances are functions that transform certain inputs concerning propositional formulas into outputs with formulas depending on the formulas that appear on the inputs.[1]

[1]There are inference rules that besides from premises, also consider open assumptions (by definition) in the input arguments. Rules like $\vee$ elimination, $\rightarrow$ introduction and $\neg$ introduction are examples of this kind. The expressive content of these rules, cannot be formulated in terms of the acceptability of *factual* arguments alone. It is exactly these rules that impose stronger conditions on how the connectives are being interpreted. Further on we will see that in every proof it is important to keep track of the propositions that were open assumptions in case they need to be dropped later by using an

We focus on the set of well-formed formulas of the language of propositional logic $FORM$.[2] Let $\Sigma_{FORM}$ denote the set of all sequences of formulas.

**Definition 18** *An* argument $(\Gamma, A)$ *is a pair in which* $\Gamma := (\Gamma^p; \Gamma^a)$ *is composed by* $\Gamma^p$ *a set and* $\Gamma^a$ *a multiset (set with repetitions) of the elements on* $FORM$*; and* $A \in FORM$ *which is, in some way to be understood later, dependent of* $\Gamma^p$ *and* $\Gamma^a$*. We will say that* $\Gamma = (\Gamma^p; \Gamma^a)$ *forms the pair that collects necessary* premises *and* open assumptions *for* $A$*. We will often refer to* $\Gamma$ *as the assumptions for* $A$*. The necessary premises will be placed in the first entry, and use ";" to indicate the separation with the second entry which will contain repetitions of the necessary open assumptions.*

**Definition 19** *A* reasoning process *is a sequence* $\mathcal{S}_0, \ldots, \mathcal{S}_n$ *for some* $n \in \mathbb{N}$*, such that for each point* $i \in \{0, \ldots, n\}$ *in the reasoning process* $\mathcal{S}_i$ *is a set of arguments, i.e.,*

$$\mathcal{S}_i := \{(\Gamma_1, A_1), (\Gamma_2, A_2), \ldots, (\Gamma_m, A_m)\}.$$

*Each* $\mathcal{S}_i$ *will be called a* reasoning stage*.*

The above signifies that conclusions $A_1, \ldots, A_m$ have been obtained from their respective assumptions $\Gamma_1, \ldots, \Gamma_m$.

**Definition 20** *Let* $r$ *be an inference rule in the usual representation. We will use* $f_r$ *to denote an instance of the rule. We will use* $[f_r]$ *to denote the set of all instances of the rule* $r$*. Thus* $r := [f_r]$*.*

Suppose $\mathcal{S}$ is a reasoning stage. We define the $R_{ND}$ rules as classes of functions taking inputs subsets of $\mathcal{S}$ and outputting arguments.

**Definition 21** $R_{ND}$ *contains the following rules:*

- *Axiom rule* $[f_{Ax}]$ *such that* $f_{AX}((\Gamma^p; \Gamma^a \cup \{\varphi\}), \varphi)) = ((\Gamma^p; \Gamma^a \cup \{\varphi\}), \varphi)$

  *Elimination rules:*

- $\wedge E^r$ *rule* $[f_{\wedge E^r}]$ *such that* $f_{\wedge E^r}((\Gamma, A \wedge B)) = (\Gamma, A)$*,*
- $\wedge E^l$ *rule* $[f_{\wedge E^l}]$ *such that* $f_{\wedge E^l}((\Gamma, A \wedge B)) = (\Gamma, B)$*,*
- $\rightarrow E$ *rule* $[f_{\rightarrow E}]$ *such that* $f_{\rightarrow E}((\Gamma, A); (\Gamma, A \rightarrow B)) = (\Gamma, B)$*,*
- $\neg E$ *rule* $[f_{\neg E}]$ *such that* $f_{\neg E}((\Gamma, A); (\Gamma, \neg A) = (\Gamma, C)$ *any* $C \in FORM$*,*
- $\vee E$ *rule* $[f_{\vee E}]$ *such that* $f_{\vee E} = (((\Gamma_1^p; \Gamma_1^a), A \vee B); ((\Gamma_2^p; \Gamma_2^a \cup \{A\}), D); ((\Gamma_3^p; \Gamma_3^a \cup \{B\}), D)) = ((\Gamma_1^p \cup \Gamma_2^p \cup \Gamma_3^p; \Gamma_1^a \cup \Gamma_2^a \cup \Gamma_3^a), D)$
- *No* $\top$ *elimination rule.*

  *Introduction rules:*

- $\wedge I$ *rule* $[f_{\wedge I}]$ *such that* $f_{\wedge I}(((\Gamma_1^p; \Gamma_1^a), A); ((\Gamma_2^p; \Gamma_2^a), B)) = ((\Gamma_1^p \cup \Gamma_2^p; \Gamma_1^a \cup \Gamma_2^a), A \wedge B)$*,*
- $\vee I^r$ *rule* $[f_{\vee I^r}]$ *such that* $f_{\vee I^r}((\Gamma, A)) = (\Gamma, A \vee B)$*,*
- $\vee I^l$ *rule* $[f_{\vee I^l}]$ *such that* $f_{\vee I^l}((\Gamma, B)) = (\Gamma, A \vee B)$*,*

appropriate rule. Our functions (rule instances) needed to account for this issue. This is precisely why we need to consider the multiset $\Gamma$ in an argument as $(\Gamma, A)$ which contains at least the necessary open assumptions for $A$, in order for $A$ to be available in the next step of the proof.

[2] In our study we are not interested in dealing with equivalent classes $\psi \leftrightarrow \phi$. Because we want our framework to address two things cognitively relevant: 1) How the order and connective-relation between assumptions and conclusions matters in a realistic reasoning/deductive process; 2) the fact that it's cognitively hard to re-arrange a given sentence using a certain connective with a different one.

- $\top I$ *rule* $[f_{\top I}]$ *such that* $f_{\top I}((\emptyset,\emptyset)) = (\emptyset, \top)$,
- $\rightarrow I$ *rule* $[f_{\rightarrow I}]$ [3] *such that* $f_{\rightarrow I}(((\Gamma^p;\Gamma^a \cup \{A\}), B)) = ((\Gamma^p;\Gamma^a), A \rightarrow B)$
- $\neg I$ *rule* $[f_{\neg I}]$ *such that* $f_{\neg I}(((\Gamma^p;\Gamma^a \cup \{A\}), C)) = ((\Gamma^p;\Gamma^a), \neg A)$ *any* $C \in FORM$.
- *No* $\perp$ *introduction rule.*

When it is clear that $A$ is an open assumption, abusing notation we will use $\Gamma \cup \{A\}$ to denote that $\Gamma^a \cup \{A\}$ is the case. The union between two multisets $\Gamma, \Gamma'$, will be executed by taking the union of premises with premises and open assumptions with open assumptions respectively. Recall that the symbol ";" appearing in $\Gamma$ serves as a differentiator between premises and open assumptions.

The rules take either one, two or three elements of a stage as inputs and the order of the input arguments does not matter.[4]

The application of the rules in these terms can be described as for $[f_{\vee E}]$.[5]. We can apply $[f_{\vee E}]$ to a reasoning stage $\mathcal{S}_i$ containing $(\Gamma_1, A \vee B); (\Gamma_2 \cup \{A\}, D); (\Gamma_3 \cup \{B\}, D)$ to obtain an extension of $S_i$, namely $\mathcal{S}_{i+1}$ which contains $(\Gamma_1 \cup \Gamma_2 \cup \Gamma_3, D)$ as the newly added element. Of course rule $[f_{\vee E}]$ might be applied to $\mathcal{S}_i$ in a different way as well if $\mathcal{S}_i$ contains other suitable formulas.

In simple words,

- $f_{\wedge E}$ works as follows: Take two available formulas $A$ and $B$ with their respective assumptions, add formula $A \wedge B$ as an available formula (available for a consecutive step in the proof) considering that $A \wedge B$ is dependent on the assumptions necessary for $A$ and $B$.
- $f_{\top I}$ works as follows: We need neither premises nor open assumptions to have truth as an available formula. Since we put no information into the proof of $\top$, we know nothing new if we have it as an assumption.
- $f_{\vee E}$ works as follows: Take an available formula $A \vee B$. Then $D$ is an available formula for a consecutive step in a proof if and only if $D$ is available when $A$ is an assumption and $D$ is also available when $B$ is an assumption.
- $f_{\neg I}$ works as follows: If when adding $A$ as an open assumption to a given set of assumptions $\Gamma$ we have any formula $C \in FORM$ as an available formula in our proof, then $A$ must prove contradiction. Thus $\neg A$ must follow as an available formula from $\Gamma$.
- Similarly for the rest of the rules.

Note that for each rule $[f_r]$ in $R_{ND}$ induces a procedure for obtaining the next reasoning stage from the previous reasoning stage in the proof.

A point worth addressing is the following: A rule is defined as a *class* of such functions to accommodate meta-variables. That said, why do we not just define inference rules as meta-functions instead? Well basically because a rule can have several outputs for a given set of inputs. This is due to the fact that the inputs for an inference rule are not governed by any specific order, but the order matters for a function. Jaśkowski's natural deduction system uses functional rules instead of natural deduction rules which get suggested by natural deduction rules; and even though it has been proven to be equivalent to Gentzen's system we could say that in principle it is a *weaker* system. In short: Natural deduction rules of inference correspond to classes of functions. But, if we use functional rules instead of natural deduction rules which get suggested by natural deduction rules, we can use weaker rules of inference that produce the same results. Gentzen's and Jaśkowski's formulations of natural deduction are logically equivalent.

[3] In order to keep some linearity in the proofs we will put the following constraint on $[f_{\rightarrow I}]$ rule: $f_{\rightarrow I}$ cannot discharge more than one assumption at the same time. So cases in which we can discharge two open assumptions which are the same proposition simultaneously are not allowed.

[4] For instance rule $[f_{\rightarrow E}]$ takes inputs of the form $(\Gamma, A)$ and $(\Gamma, A \rightarrow B)$, which can be taken in two different orders: $f_{\rightarrow E}((\Gamma, A), (\Gamma, A \rightarrow B))$ and $f_{\rightarrow E}((\Gamma, A \rightarrow B), (\Gamma, A))$.

[5] The application of the rule allows to use it for the maximal set of necessary assumptions. For instance in $[f_{\vee E}]$: We can apply this rule in the following manner,

$$f_{\vee E}((\Gamma_1 \cup \Gamma_2 \cup \Gamma_3, A \vee B); (\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \{A\}, D); (\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \{B\}, D)) = (\Gamma_1 \cup \Gamma_2 \cup \Gamma_3, D).$$

However, Gentzen's formulation more straightforwardly lends itself both to a normalization theorem and to a theory of "meaning" for connectives. In (42 - Pelletier and Hazen), the authors investigate cases where Jaśkowski's formulation seems better suited. These cases range from the phenomenology and epistemology of proof construction to the ways to incorporate novel logical connectives into the language.

### 4.2.1 Proofs corresponding to $R_{ND}$

Now we need to address what is a *proof* in the system $R_{ND}$. In a very general way, proofs will be sequences of reasoning stages in which each element of the sequence was obtained by rule application of some rule in $R_{ND}$.

In order to illustrate $R_{ND}$-proofs, consider the following example:

**Example 5** *Take $P$ to be a $R_{ND}$-proof of $A \to (B \to (A \wedge B))$ starting with $\emptyset$ premises and open assumptions $A$, $B$. $P$ is a sequence of stages of conclusions $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ such that:*
$\mathcal{S}_0 = \{((\emptyset; A \cup B), A), ((\emptyset; A \cup B), B)\}$,
$\mathcal{S}_1 = \{((\emptyset; A \cup B), A), ((\emptyset; A \cup B), B), ((\emptyset; A \cup B), A \wedge B)\}$,
$\mathcal{S}_2 = \{((\emptyset; A \cup B), A), ((\emptyset; A \cup B), B), ((\emptyset; A \cup B), A \wedge B), ((\emptyset; A), B \to (A \wedge B))\}$,
$\mathcal{S}_3 = \{((\emptyset; A \cup B), A), ((\emptyset; A \cup B), B), ((\emptyset; A \cup B), A \wedge B), ((\emptyset; A), B \to (A \wedge B)), ((\emptyset; \emptyset), A \to (B \to (A \wedge B)))\}$,

*and $((\emptyset; \emptyset), A \to (B \to (A \wedge B))) \in \mathcal{S}_3$.*

*The composition of rule instances goes as $f_{\to I}(f_{\to I}(f_{\wedge I}(((\emptyset; A \cup B), A), ((\emptyset; A \cup B), B))))$ which outputs $(\emptyset, A \to (B \to (A \wedge B)))$. The process goes as follows: First take $f_{\wedge I}$ with input $((\emptyset; A \cup B), A), ((\emptyset; A \cup B), B))$ which outputs $((\emptyset; A \cup B), A \wedge B)$, then we take $f_{\to I}$ with input $((\emptyset; A \cup B), A \wedge B)$ which outputs $((\emptyset; A), B \to (A \wedge B))$. Observe that open assumption $B$ was deleted from the left side of the pair. Finally we take $f_{\to I}$ with input $((\emptyset; A), B \to (A \wedge B))$ which outputs $((\emptyset; \emptyset), A \to (B \to (A \wedge B)))$.*

Notation: We will use $f_r : \mathcal{S}$ to label the rule which was applied for obtaining stage $\mathcal{S}$.

Example 5 suggests that the proofs as sequences of reasoning stages behave similarly to the compositions of functions representing the instances of the rules. We make it precise with the following definition.

**Definition 22** *Let $\Gamma$ be a pair $(\Gamma, \Gamma')$ of finite multisets of elements in $FORM$ and $C \in FORM$. We call $P_{\Gamma,C}$ a $R_{ND}$-proof of $C$ with $\Gamma$ assumptions (premises and necessary open assumption), if $P_{\Gamma,C}$ is a reasoning process $\mathcal{S}_0, \ldots, \mathcal{S}_n$ where for each $i \in \{1, \ldots, n\}$, $S_i$ is obtained from the previous one according to an application of some rule $[f_r] \in R_{ND}$; $(\Gamma, C) \in \mathcal{S}_n$; and $(\Gamma, C) \notin \mathcal{S}_k$ for any $k \in \{0, \ldots, n-1\}$.*[6]

**Definition 23** *$\langle R_{ND} \rangle$ is the set of all $R_{ND}$-proofs.*

We will consider a more informative proof representation, which can be called *labelled proofs*. The most informative labels we will consider in a proof will be the rules used at each reasoning stage with their corresponding inputs. We will refer to this labeled proof a *complete labeled proof*. A more relaxed way of labeling a proof is with the inputs taken by the rule which performed the corresponding stage extension. We will refer to this relaxed labeled proof *partial labelled proof*. To illustrate how a labelled proof looks like, consider Example 5: A labelled proof in the most informative way will be: $\mathcal{S}_0, f^{1,2}_{\wedge I} : \mathcal{S}_1, f^{2,3}_{\to I} : \mathcal{S}_2, f^{1,4}_{\to I} : \mathcal{S}_3$ such that each reasoning stage is enumerated as,
$\mathcal{S}_0 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2\}$,
$\mathcal{S}_1 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3\}$,
$\mathcal{S}_2 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3, ((\emptyset; A), B \to (A \wedge B))_4\}$,
$\mathcal{S}_3 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3, ((\emptyset; A), B \to (A \wedge B))_4, ((\emptyset; \emptyset), A \to (B \to (A \wedge B)))_5\}$,

[6] $R_{ND}$-proofs are ordered sets which supports the view in theory of reasoning saying that deductive reasoning processes follow an order. Secondly, they are local, so that all information that will be needed in future steps must be carried along up to that point. A computational advantage of having this representation for $R_{ND}$-proofs in terms of *time efficiency* is that the learner only needs to consider the previous stage in the proof in order to build the next stage of the proof.

and $((\emptyset;\emptyset), A \to (B \to (A \wedge B))) \in \mathcal{S}_3$.
A partial labeled proof will be $\mathcal{S}_0, (1,2) : \mathcal{S}_1, (2,3) : \mathcal{S}_2, (1,4) : \mathcal{S}_3$ such that each reasoning stage is as before.

### 4.2.2 The correspondence between natural deduction and $R_{ND}$

It is possible to mimic the rule usage of a natural deduction proof $R_{ND}$. First we will provide a procedure for constructing a $R_{ND}$-proof from a natural deduction proof (in the linear representation). Second, we will show the relation between ND and $R_{ND}$ by giving a sketch for a proof of the correspondence between these two.

Given a proof $\Omega$ for $y$ in natural deduction we can construct each reasoning stage for a $R_{ND}$-proof with the following procedure.[7]

Let $\Gamma := \{\gamma_1, \ldots, \gamma_k\}$ be the set containing all premises and open assumptions that appear in $\Omega$.

- Let $\mathcal{S}_0 := \{(\Gamma, \gamma_0), \ldots, (\Gamma, \gamma_k)\}$;
- Take the first inference $y_1 \in FORM$ that was made from taking at most three elements $\gamma', \gamma'', \gamma'''$ in $\Gamma$ by the application of a rule $r \in ND$ in $\Omega$ and let $\mathcal{S}_1 := \{(\Gamma, \gamma_0), \ldots, (\Gamma, \gamma_k), (\Gamma, y_1)\}$. Note that $\mathcal{S}_1$ is an extension of $\mathcal{S}_0$ and $(\Gamma, y_1)$ is the output of rule $[f_r]$ when $f_r$ takes $(\Gamma, \gamma'), (\Gamma, \gamma''), (\Gamma, \gamma''')$ (at most) as inputs from $\mathcal{S}_0$.
- Take the next inference $y_2 \in FORM$ that was made after inference $y_1$ in $\Omega$ and repeat the process above but for constructing $\mathcal{S}_2$ taking inputs in $\mathcal{S}_1$ instead.
- Continue repeating the process above for every $y_i \in FORM$ that appeared in $\Omega$ until we reach the desired conclusion $y_n = y$ (the root in a tree proof or the final element in the list of a linear proof).
- The final reasoning stage will be $\mathcal{S}_n$ such that it contains $(\Gamma, y)$ and was constructed by using some rule $[f_r] \in \mathcal{R}_{ND}$ which corresponds to a certain rule $r$ in ND taking inputs in $\mathcal{S}_{n-1}$.

The procedure described above, allows to translate any proof in ND into a $R_{ND}$-proof which provides a proof for the following proposition.

**Proposition 1** *Any natural deduction proof can be rewritten as a $R_{ND}$-proof.* □

In the following proposition we can observe a sketchy construction of a proof for the other direction of the correspondence.

**Proposition 2** *Any $R_{ND}$-proof can be rewritten as a natural deduction proof.*

*Proof:* This is by induction over the last rule applied on the previous reasoning stage in the reasoning process; i.e. over the outermost function application when we *think* of $R_{ND}$-proofs as composition of functions corresponding to the rules that were used. There are several cases:

1. The last function applied was $f_{AX}(\Gamma \cup \{\varphi\}, \varphi) = (\Gamma \cup \{\varphi\}, \varphi)$ then the proof in classical propositional logic ends in $\Gamma, \varphi \vdash \varphi$.
2. The last function applied was $f_{\to E}$ to a previous stage of conclusions $\mathcal{S}$ in the reasoning process containing the pair $(\Gamma, \varphi \to \psi)$ and $(\Gamma, \varphi)$ such that $f_{\to E}((\Gamma, \varphi \to \psi); (\Gamma, \varphi)) = (\Gamma, \psi)$. By induction hypothesis we have $\Gamma \vdash \varphi \to \psi$ and $\Gamma \vdash \varphi$. Thus,

   $$\frac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

   is a ND proof in classical logic.

[7] It can be done either in the linear representation and be easily adapted to the tree representation since there is a well-known correspondence between linear proofs and tree proofs ((54, 3), also note that we did not put any constraint in our definition of $R_{ND}$-proofs so they can be adapted to both linear and tree proofs.

3. The last function applied was $f_{\to I}$ to a previous reasoning stage $\mathcal{S}$ in the reasoning process containing $(\Gamma \cup \{\varphi\}, \psi)$ such that $f_{\to I}((\Gamma \cup \{\varphi\}, \psi)) = (\Gamma, \varphi \to \psi)$. By induction hypothesis, we obtain $\Gamma, \varphi \vdash \psi$. Thus,

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}$$

is a ND proof in classical logic.

4. The last function applied was $f_{\wedge I}$ to a previous reasoning stage $\mathcal{S}$ in the reasoning process containing $(\Gamma, \varphi)$ and $(\Gamma, \psi)$ such that $f_{\wedge}((\Gamma, \varphi); (\Gamma, \psi)) = (\Gamma, \varphi \wedge \psi)$. By induction hypothesis we have $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi$. Thus,

$$\frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}$$

is a ND proof in classical logic.

5. The last function applied was $f_{\wedge E^r}$ to a previous reasoning stage $\mathcal{S}$ in the reasoning process containing $(\Gamma, \varphi \wedge \psi)$ such that $f_{\wedge E^r}((\Gamma, \varphi \wedge \psi)) = (\Gamma, \varphi)$. By induction hypothesis we obtain $\Gamma \vdash \varphi \wedge \psi$. Thus,

$$\frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi}$$

is a ND proof in classical logic.

6. The last function applied was $f_{\wedge E^l}$ to a previous reasoning stage $\mathcal{S}$ in the reasoning process containing $(\Gamma, \varphi \wedge \psi)$, analogously as the previous case it is easy to see that

$$\frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}$$

is a ND proof in classical logic.

7. The last function applied is $f_{\vee I^r}$ to a previous reasoning stage $\mathcal{S}$ in the reasoning process containing $(\Gamma, \varphi)$ such that $f_{\vee I^r}((\Gamma, \varphi)) = (\Gamma, \varphi \vee \psi)$. By induction hypothesis we have $\Gamma \vdash \varphi$. Thus,

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi}$$

is a ND proof in classical logic.

8. Similar for $f_{\vee I^l}$, we obtain that

$$\frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}$$

is a ND proof in classical logic.

9. The last function applied is $f_{\vee E}$ to a previous reasoning stage $\mathcal{S}$ in the reasoning process containing $(\Gamma, \varphi \vee \psi)$, $(\Gamma \cup \{\varphi\}, \beta)$ when $\{\varphi\}$ is an open assumption, and $(\Gamma \cup \{\psi\}, \beta)$ when $\{\psi\}$ is an open assumption; such that $f_{\vee E}((\Gamma, \varphi \vee \psi); (\Gamma \cup \{\varphi\}, \beta); (\Gamma \cup \{\psi\}, \beta)$ outputs $(\Gamma, \beta)$. By induction hypothesis we obtain $\Gamma \vdash \varphi \vee \psi$, $\Gamma, \varphi \vdash \beta$ and $\Gamma, \psi \vdash \beta$ considering open assumptions $\varphi$, $\psi$ . Thus,

$$\frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \beta \quad \Gamma, \psi \vdash \beta}{\Gamma \vdash \beta}$$ is a ND proof in classical logic. □

## 4.3 The class of alternative inference systems

Let us now discuss the class of possible reasoning systems the learner chooses the hypothesis from. The class should in some way generalize natural deduction reasoning system. The first intuition could be that the class should consists of systems composed of functions transforming premises into conclusions in ways similar to those of natural deduction.

**Definition 24** *We will use the following notation.*

- *Let $\mathcal{F}$ denote the family of functions of the form*

$$f : (\Sigma_{FORM}, FORM)^{\leq m} \to (\Sigma_{FORM}, FORM)^{\leq n} \tag{4.1}$$

  *for every $m, n \in \mathbb{N}$.*

- *Let $\mathcal{F}^{\leq 3}$ denote the family of functions of the form*

$$f : (\Sigma_{FORM}, FORM)^{\leq 3} \to (\Sigma_{FORM}, FORM) \tag{4.2}$$

Inference rules can be understood in such a way, having arbitrary size of inputs and arbitrary size of outputs as appears in 4.1 of Definition 24 . It seems to be too much to assume that the class $R$ should include systems defined over arbitrary elements of $R$, after all it is hard to find logical operators which take more than 2 arguments.

**Definition 25** *An* inference system *is a set of rules. A rule is a set of* rule instances. *A* rule instance *is a function that transforms a (possibly empty) set of arguments (called the inputs) into a new argument (called the output).*

We will hence assume the alternative systems to be as in Definition 25; moreover that the elements of $\mathcal{R}$ include systems over $\mathcal{F}^{\leq 3}$. Even under this restriction, the class $\mathcal{R}$ containing all systems over functions from $\mathcal{F}^{\leq 3}$ seems to be too large. We will hence decide to consider a class $R$ of systems "as similar as possible" to $R_{ND}$. This similarity will be obtained by assuming that the systems alternative to $R_{ND}$ consist of misassigned ND-rules, i.e, systems in which the rules of reasoning are present, but logical connectives are misinterpreted.

**Definition 26** *We will make use of the following notation:*

- *$R^T$ will denote the system in $\mathcal{R}$ that needs to be learned. We will often refer to $R^T$ as* the target set. *Note that we are interested in learning not only $R_{ND}$ but the whole class $\mathcal{R}$, thus $R^T$ will not always refer to $R_{ND}$.*

- *$\langle R \rangle$ will denote the set generated by $R$. By this we mean the set of all $R$-proofs obtained by using correctly the inference rules in $R$ starting with some axioms. We will often call $\langle R \rangle$ the* language *of $R$.*

We will intend the relation between inference systems $R$ and the set of corresponding complete proofs $\langle R \rangle$ to be as the one between grammars and languages described in Chapters 2 and Section 3.3 in Chapter 3.

### 4.3.1 Inference rules and their functional character

As in $R_{ND}$, the rules of inference in the alternative systems will also have a functional character. An $R$-proof will be defined is the same way as for $R_{ND}$. That said, an inference rule $[f_r]$ will take elements of a reasoning stage $\mathcal{S}_i$ as inputs and will output new arguments which will serve to extend the current reasoning stage to form $\mathcal{S}_{i+1}$; and these new arguments will serve as inputs for future reasoning stages if we continue the reasoning process.

For our learning class $\mathcal{R}$, we will only be interested in inference systems whose rules are formed as misinterpretations between the inputs of some natural deduction rule (ND-rule) and the output from that or another ND rule.

### 4.3.2 The alternative inference systems

For constructing the rest of the inference systems in $\mathcal{R}$ we first define a family $\mathbb{B}$ of sets such that each member of this family will be associated to the respective natural deduction rule.

**Definition 27** *We define the following sets for each ND-rule:*

- *Consider $\mathcal{B}_{\wedge I}$ to be the family of rules (classes of functions) $[f_r]$ of which the instances are either of the form*

$$f_r((\Gamma_1, A), (\Gamma_2, B)) = (\Gamma_3, X \bowtie Y)$$

*such that $X, Y \in \{A, B\}$ and $\bowtie \in \{\wedge, \vee, \rightarrow\}$ and $\Gamma_3$ is dependent on $\Gamma_1, \Gamma_2$,*
*or of the form,*

$$f_r((\Gamma_1, A), (\Gamma_2, B)) = (\Gamma_3, \neg X)$$

*such that $X \in \{A, B\}$ and $\Gamma_3$ is dependent on $\Gamma_1, \Gamma_2$.*

  - *In simple words, the elements of $\mathcal{B}_{\wedge I}$ are classes of functions similar to $f_{\wedge I}$. So each class $[f_r] \in \mathcal{B}_{\wedge I}$ is a possible combination of arguments as in $[f_{\wedge I}]$ (based on $A$, $B$ and their respective open assumptions) with outputs based on $A$, $B$ and a connective.*
  - *Clearly $[f_{\wedge I}] \in \mathcal{B}_{\wedge I}$.*

- *Consider $\mathcal{B}_{\rightarrow I}$ to be the family of rules (classes of functions) $[f_r]$ of which the instances are either of the form*

$$f_r((\Gamma_1 \cup A), B)) = (\Gamma_2, X \bowtie Y)$$

*such that $X, Y \in \{A, B\}$ and $\bowtie \in \{\wedge, \vee, \rightarrow\}$ and $\Gamma_2$ is dependent on $\Gamma_1$ or of the form,*

$$f_r((\Gamma_1 \cup A), B)) = (\Gamma_2, \neg X)$$

*such that $X \in \{A, B\}$ and $\Gamma_2$ is dependent on $\Gamma_1$.*

- *Consider $\mathcal{B}_{\wedge E} = \mathcal{B}^l_{\wedge E} \cup \mathcal{B}^r_{\wedge E}$ to be such that $\mathcal{B}^l_{\wedge E}$ is the family of classes of functions which have either the form*

$$f((\Gamma_1, A \bowtie B)) = (\Gamma_2, X)$$

*such that $X \in \{A, B\}$, $\bowtie \in \{\wedge, \vee, \rightarrow\}$ and $\Gamma_2$ depends on $\Gamma_1$,*
*or the form,*

$$f((\Gamma_1, A \bowtie B)) = (\Gamma_2, \neg X)$$

*such that $X \in \{A, B\}$, $\bowtie \in \{\wedge, \vee, \rightarrow\}$ and $\Gamma_2$ depends on $\Gamma_1$ Similarly for $\mathcal{B}^r_{\wedge E}$. Thus $\mathcal{B}_{\wedge E} = \mathcal{B}^l_{\wedge E} \cup \mathcal{B}^r_{\wedge E}$ has the possible combinations for arguments based on $A, B$ and a connective, with outputs based on $A$ or $B$. Observe that $f_{\wedge E^l}, f_{\wedge E^r} \in \mathcal{B}_{\wedge E}$.*

- *Consider $\mathcal{B}_{\vee E}$ [8] to be the family of classes of functions which have either the form*

$$f((\Gamma_1, A \bowtie B); (\Gamma_2 \cup \{A\}, C); (\Gamma_3 \cup \{B\}, C)) = (\Gamma_4, X)$$

*where $X \in \{A, B, C\}$, $\bowtie \in \{\wedge, \vee, \rightarrow\}$ and $\Gamma_4$ depends on $\Gamma_1, \Gamma_2, \Gamma_3, \{A\}$ and $\{B\}$,*
*or the form,*

$$f((\Gamma_1, A \bowtie B); (\Gamma_2 \cup \{A\}, C); (\Gamma_3 \cup \{B\}, C)) = (\Gamma_4, \neg X)$$

*where $X \in \{A, B, C\}$, $\bowtie \in \{\wedge, \vee, \rightarrow\}$ and $\Gamma_4$ depends on $\Gamma_1, \Gamma_2, \Gamma_3, \{A\}$ and $\{B\}$.*

*In a similar fashion we obtain the sets $\mathcal{B}_{\vee I} = \mathcal{B}^l_{\vee I} \cup \mathcal{B}^r_{\vee I}$; $\mathcal{B}_{\vee E}$; $\mathcal{B}_{\rightarrow E}$; $\mathcal{B}_{\neg I}$ and $\mathcal{B}_{\neg E}$.*

[8] Note that for $\mathcal{B}_{\vee E}$ there is a proposition $C$ playing an important role, the necessary assumptions for $C$ depend also on the necessary assumptions for $A$ and $B$. A similar thing happens for $\mathcal{B}_{\rightarrow I}$, $\mathcal{B}_{\neg E}$ and for $\mathcal{B}_{\neg E}$.

Now, let $\mathbb{B}$ be the following set

$$\mathbb{B} := \{\mathcal{B}_{\wedge I}, \mathcal{B}^r_{\wedge E}, \mathcal{B}^l_{\wedge E}, \mathcal{B}^r_{\vee I}, \mathcal{B}^l_{\vee I}, \mathcal{B}_{\vee E}, \mathcal{B}_{\to E}, \mathcal{B}_{\to I}, \mathcal{B}_{\neg I}, \mathcal{B}_{\neg E}\}.$$

For instance, take rule $[f_r] \in \mathcal{B}_{\wedge I}$ such that the instances of the rule are of the form $f_r((\Gamma_1, A); (\Gamma_2, B)) = (\Gamma_3, A \vee B)$. Observe that $[f_r] \neq [f_{\wedge I}]$. In this case the confusion is between connectives $AND$ and $OR$.

Note that $R_{ND}$ can be *chosen* from $\mathbb{B}$ by taking exactly one appropriate element from each $\mathcal{B} \in \mathbb{B}$.

**Observation 1** *There is a finite number of natural deduction rules, thus the number of possible forms of conclusions is finite. Also in each natural deduction rule, the number of arguments as inputs is finite (at most three). Thus we have that the number of possible ways of combining the inputs with the conclusions(outputs) is finite. Therefore each $\mathcal{B} \in \mathbb{B}$ is finite. Therefore, each $\mathcal{B}$ is finite by construction.*

In the following definition, we can observe that abusing our notation the class $\mathcal{R}$ is a subset of $\mathcal{P}(\mathcal{F}^{\leq 3})$ bounded by $\mathbb{B}$.

**Definition 28** *An* inference system $R \in \mathcal{R}$ *is a set of the following form,*

$$[f_r] \in R \text{ iff } [f_r] \in \bigcup \mathbb{B}$$

*Thus, $\mathcal{R}$ is the family of all these sets $R$. Let $I_{\mathcal{R}}$ be the set of indices for $\mathcal{R}$.*

Class $\mathcal{R}$ contains the special subclass $\mathcal{R}^{\leq 1}$ in which $R \cap \mathcal{B}_j \leq 1$ for each $\mathcal{B}_j \in \mathbb{B}$ holds for every $R \in \mathcal{R}^{\leq 1}$. This means that at most one misinterpretation for each natural deduction rule is allowed. We will call such types of inference systems *deterministic*. We wanted our learning space to at least contain such subclass. This subclass contains usual cases in which people identify each connective as different from the rest of the connectives but do not know the correct interpretation. However, class $\mathcal{R}$ can have inference systems in which more than one interpretation is assign to one connective, so cases in which people misinterpret $\wedge$ with $\vee$ in some context but also interpret $\wedge$ correctly as $\wedge$ in a different context are systems we wanted to be considered in $\mathcal{R}$ and for similar cases considering other connectives. For instance consider $R \in \mathcal{R}$ such that $R$ contains rule $[f^r_{\wedge I}]$ and also contains rule $[f_1]$ in which $f_1((\Gamma, A); (\Gamma, B)) = (\Gamma, A \vee B)$. So an agent with such system can sometimes interpret $AND$ as an $OR$. We will call $\mathcal{R}^{\geq 1}$ to such subclass, in which $R \cap \mathcal{B}_j \geq 1$ for each $\mathcal{B}_j \in \mathbb{B}$ holds for every $R \in \mathcal{R}^{\geq 1}$. We will call such types of inference systems *all-inclusive*. If this is the case, it seems natural to impose certain probability distribution over the possible interpretations allowed in the same system. For instance the reasoner might be more likely to treat conjunction as a disjunction than as an implication. It is also likely that the actual interpretations of connectives depend on the context in which they occur (53). However we will not implement probabilities in this work. In a way one may think of *tableaux inference system* style as the one that is precisely deterministic, elimination-exhaustive (with the appropriate rules) and introduction-empty (no introduction rules).

$R \neq R'$ for every $R, R'$ since we are dealing with possible *combinations* i.e., permutations without repetition. There are inference systems which are exactly as $R_{ND}$ but were formed by confusing $[f^r_{\wedge E}]$ with $[f^l_{\wedge E}]$ and $[f^l_{\wedge E}]$ with $[f^r_{\wedge E}]$; and/or by confusing $[f^l_{\vee I}]$ with $[f^r_{\vee I}]$ and $[f^r_{\vee I}]$ with $[f^l_{\vee I}]$. In principle, these inference systems are "different" from $R_{ND}$ however they contain exactly the same classes of functions as $R_{ND}$. As a matter of fact, $[f_r] \in R_{ND}$ if and only if $[f_r] \in R$ when $R$ is one of these inference systems that confuse *left* with *right*. Therefore we will treat this systems as one which will be $R_{ND}$. Also for similar cases.

**Observation 2** *$\mathcal{R}$ is finite. This is because since each $\mathcal{B} \in \mathbb{B}$ is finite and $\mathbb{B}$ is finite, so we can have only finitely many combinations of the elements of $\mathcal{B}$'s in $\mathbb{B}$.*

What is the relation between $R, R'$ in $\mathcal{R}$? We can have cases in which $R \subseteq R'$, $R \cap R' = \emptyset$ and $R \cap R' \neq \emptyset$. Take the following examples:

- Take the rules $[f_1]$ such that $f_1((\Gamma_1, A \wedge B)) = (\Gamma_2, A)$, $[f_2]$ such that $f_2((\Gamma'_1, A \wedge B)) = (\Gamma'_2, \neg B)$ which come from $\mathcal{B}_{\wedge E}$ and $[f_3]$ such that $f_3((\Gamma_1, A); (\Gamma_2, B)) = (\Gamma_3, A \vee B)$ which comes from $\mathcal{B}_{\wedge I}$. Take the inference sets $R := \{[f_1], [f_3]\}$ and $R' := \{[f_1], [f_2], [f_3]\}$. Clearly $R, R' \in \mathcal{R}$ and $R \subseteq R'$.

- Take $R_{ND}$ and $R = (R_{ND} \setminus \{[f_{\wedge I}]\}) \cup \{[f_3]\}$ such that $[f_3]$ takes two available formulas $A$ and $B$ with their respective assumptions; and outputs $A \vee B$ with the respective assumptions. Then $R_{ND} \cap R \neq \emptyset$.
- Take $R$ to be the set containing only the introduction rules of $R_{ND}$ and $R'$ to be the set containing only the elimination rules of $R_{ND}$. Then $R \cap R' = \emptyset$.

### 4.3.3 The proofs of an alternative inference system

What are "proofs" using an inference system $R_i$? As for $R_{ND}$, $R_i$-proofs are sequences of stages of conclusions in which each element of the sequence was obtained by rule application of some rule in $R_i$. These sequences of stages of conclusions $\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_k$ can be labeled as we mentioned for $R_{ND}$, i.e., with the rules and/or inputs that were used to extend each stage.

The proofs as sequences of stages have a similar behaviour as compositions of functions representing the rules. We will see in the following definition that to some extent they are.

**Definition 29** *Let $R_i \in \mathcal{R}$, $\Gamma$ a finite multiset of propositional formula and $C$ a propositional formula. We call $P_{\Gamma,C}$ a $R_i$-proof of $C$ with $\Gamma$ assumptions (premises and necessary open assumption), if $P_{\Gamma,C}$ is a finite sequence $\mathcal{S}_0, \ldots, \mathcal{S}_n$ of stages of conclusions where each one is obtained from the previous one according to a rule application for some rule $[f_r] \in R$; $(\Gamma, C) \in \mathcal{S}_n$; and $(\Gamma, C) \notin \mathcal{S}_k$ for any $k \in \{0, \ldots, n-1\}$.*

**Definition 30** *$\langle R_i \rangle$ is the set of $R_i$-proofs for every $i \in I_{\mathcal{R}}$ .*

**Proposition 3** *R-proofs are ordered sets.*

*Proof:* Straightforward by definition and the usual subset relation ($\subseteq$) over the reasoning stages. □
Proposition 3 supports the view in theory of reasoning saying that reasoning processes follow an order (8). Secondly, they are local, so that all information that will be needed in future steps must be carried along up to that point.

## 4.4 Conclusion

In this chapter we focused on formalizing mathematically the objects to be learned. We define formally the rules which will play the role of misinterpretations on the rules in natural deduction; such rules will compose the alternative inference systems. We needed to be very formal, explicit and precise on how these erroneous inference systems were. This is because the inference systems will affect directly the "mathematical structure and shape" of the learning space under consideration; and having a clear picture of the mathematical characteristics of the learning space is necessary for choosing an appropriate learnability framework.

# Chapter 5

# Learnability of inference systems

## 5.1 Introduction

In some formal learning theory frameworks, learning something requires more than just to declare to have learned it. In order to claim that an agent actually *learns that* $\varphi$ we sometimes require that the agent got to *know* that $\varphi$. Our framework relies on a weaker notion of learning since it only requires that the agent eventually converges to the accurate concept, even without realizing that he *knows* the concept. In any case one needs to provide some insights into the way in which the incoming information is presented, and into their relation with the concepts to be learned. Usually, the pieces of data are of a different, simpler nature than the concept being learned and the data stream is available over the "learning process" over more than one step. Note that it is very important to select an appropriate stream of data since these pieces of information should be the "clues" that will lead the agent to learn $\varphi$. The relationship between data and hypothesis should be like the one between sentences and grammars. Also, the learning process can go either unsupervised or supervised by means of a *teacher*. In a supervised environment, it is expected of the teacher that she, every once in a while, makes some interventions in the learning process.

Recall that a formal learning model consists of: the class being learned (class $\mathcal{R}$ from Chapter 4); a definition of learnability (identification in the limit from Chapter 3); as well as a method of information presentation and a specification of the learning function. The last two components will be addressed in the present chapter. Regarding the method of data presentation, we will consider two main types: complete proofs and, in various ways, incomplete arguments. These, will have different implications in an unsupervised learning process and in a supervised one.

One can easily agree that in real life there are different ways in which we receive information depending on the situation or environment we are in. Take the case of a trial in which the judge receives complete step by step argumentation (supported with the respective evidence) from both parties. But how was the *way* in which the detective in charge of the investigation received the relevant information for him to conclude something? Probably, he had some premises and maybe some open assumptions (based on some evidence) but nothing more. He needed to re-construct the *sequence of steps* of the case, to analyze what really happened. Therefore, the judge received complete information while the detective received partial information. This simple example illustrates how sometimes in real life reasoning information can be presented in different forms; sometimes less and sometimes more informative. Another example is when an expert $L$ in a certain field presents some conjecture to another colleague $T$ (also an expert in that same field); it is possible that $L$ receives a negative answer from $T$ and, as expected, such negative reaction should be accompanied by a counterexample.

We will go into details soon, but for now let us use a general symbol $\Pi$ to represent the set of all data items for defining our learning function.

**Definition 31** *The learning function $L$ is a map from finite sequences of data, $\Pi^*$, into inference systems in $\mathcal{R}$, i.e.,*

$$L : \Pi^* \to \mathcal{R}$$

## 5.2 How to present the data

The stream of data will consist of data items presented as pairs $(\hat{x}, y)$ in which the reasoning $\hat{x}$ leads to conclusion $y$; and the sequence $\hat{x}$ will take five different forms. Our streams of data will always fully represent the underlying inference systems, i.e., they are: *truthful* so the agent receives only correct data; and *complete* so that in the long run, all information will be provided.

1. Truthfulness (soundness): The agent receives only true data, no false information.
2. Completeness: Full stream of data is available; this means that in the long run, full information will be provided.

We will focus on two main ways of the presentation of positive data: complete proofs and reduced proofs. The complete proofs will be such that $\hat{x}$ is a sequence of reasoning stages $\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_n$, $y \in \mathcal{S}_n$ but $y \notin \mathcal{S}_k$ for $k \in \{0, \ldots, n-1\}$. They can be ordered in a way in which which $\mathcal{S}_n$ is an extension of the previous ones. For the next step in the reasoning process $\mathcal{S}_n$ will provide inputs for an inference rule that will extend $\mathcal{S}_n$ with the given outputs to become $\mathcal{S}_{n+1}$ and so on. Reduced proofs expressed as pairs $(\hat{x}, y)$ will corresponds to *incomplete* proofs; either missing reasoning stages or by presenting just assumptions $\hat{x}$ (set of premises and open assumptions) for $y$. Further on we will see three different methods of presenting information as complete proofs and two different methods of presenting information as reduced proofs as summarized in the following table.

| **Proofs** | **Method of information** |
|---|---|
| Complete | 1) Rule and input specification.<br>2) Input specification<br>3) Proof sequence |
| Reduced | 4) Initial and last reasoning stages<br>5) Set of premises and open assumptions |

Figure 5.1: Proofs and methods of information.

To illustrate these forms consider the following example.

**Example 6** *The formula $A \to B \to (A \wedge B)$ can be proved by $R_{ND}$; starting with empty premises and $A$, $B$ as open assumptions (which will be dropped later by means of $[f_{\to I}]$). We can present* this *claim in five different forms, each less informative than the previous one.*

1. *Fully labelled proof: $(\hat{x}, y)$ where*

   - $\hat{x} = \mathcal{S}_0, f^{1,2}_{\wedge I} : \mathcal{S}_1, f^{2,3}_{\to I} : \mathcal{S}_2, f^{1,4}_{\to I} : \mathcal{S}_3$ *where*

     $\mathcal{S}_0 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2\}$,
     $\mathcal{S}_1 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3\}$,
     $\mathcal{S}_2 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3, ((\emptyset; A), B \to (A \wedge B))_4\}$,
     $\mathcal{S}_3 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3, ((\emptyset; A), B \to (A \wedge B))_4, ((\emptyset; \emptyset), A \to (B \to (A \wedge B)))_5\}$

   - $y = ((\emptyset; \emptyset), A \to (B \to (A \wedge B))) \in \mathcal{S}_3$.

     *Recall that we use symbol ";" in the multiset of assumptions $\Gamma$ to separate premises from open assumptions where the premises are placed on the left side of ";".*

2. *Partially labelled proof: $(\hat{x}, y)$ where*

   - $\hat{x} = \mathcal{S}_0, (1,2) : \mathcal{S}_1, (2,3) : \mathcal{S}_2, (1,4) : \mathcal{S}_3$ *where*

     $\mathcal{S}_0 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2\}$,
     $\mathcal{S}_1 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3\}$,

$\mathcal{S}_2 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3, ((\emptyset; A), B \to (A \wedge B))_4\}$,
$\mathcal{S}_3 = \{((\emptyset; A \cup B), A)_1, ((\emptyset; A \cup B), B)_2, ((\emptyset; A \cup B), A \wedge B)_3, ((\emptyset; A), B \to (A \wedge B))_4, ((\emptyset; \emptyset), A \to (B \to (A \wedge B)))_5\}$

- $y = ((\emptyset; \emptyset), A \to (B \to (A \wedge B))) \in \mathcal{S}_3$.
  *In this case, the inputs that were taken by the rules to extend each stage where specified but not the rules.*

3. *Non-labelled complete proof:* $(\hat{x}, y)$ *where*

   - $\hat{x} = \mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ *where* $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ *are as before.*
   - $y = ((\emptyset; \emptyset), A \to (B \to (A \wedge B))) \in \mathcal{S}_3$.
     *In this case the proof sequence without any specification is presented.*

4. *First and last reasoning stages:* $(\hat{x}, y)$ *where:*

   - $\hat{x} = \mathcal{S}_0, \mathcal{S}_3$ *where* $\mathcal{S}_0, \mathcal{S}_3$ *are as before but its elements are not enumerated.*
   - $y = ((\emptyset; \emptyset), A \to (B \to (A \wedge B))) \in \mathcal{S}_3$.
     *In this case, only the initial reasoning stage and the final stage of the proof sequence are presented; the final stage can be shuffled.*

5. *Set of premises and open assumption:* $(\hat{x}, y)$ *where* $\hat{x} = \{\emptyset; A \cup B\}$ *is simply the set of premises and open assumptions used to construct the proof sequence and* $y = A \to B \to (A \wedge B)$.

You can observe that the main changes on $(\hat{x}, y)$'s presentation depend on $\hat{x}$. That said, the sequence $\hat{x}$ will take one of the following forms (see the respective enumeration in Example 6):

1. Fully labelled proof: A sequence of reasoning stages with the specific rules that were used to obtain them. We could think of it as a classroom scenario in which deductive proofs are presented step by step with all the information concerning the proof (rules with the specific application, premises, open assumptions, order etc).

2. Partially labelled proof: A sequence of reasoning stages without the specific rules that were used but specifying the inputs they took to extend each stage. We could think again of a classroom scenario, now one in which the learner is guessing which rules where used in each step of a proof. The learner evaluates if the rules of the inference system being learned fit in the proof steps with the respective inputs.

3. Non-labelled complete proof: A sequence of reasoning stages without any extra specification. Now we could think of the trial scenario for the case of the judge described before, where complete sequences of reasoning stages (according to evidence) where presented during a trial. Considering the *classroom* environment again, the proof is not explicitly given to the learner. Now he might be guessing which rules where used in each step of a proof and which inputs (either premises, open assumptions or previous inferences) were considered at each stage in the proof to obtain the next stage and so on.

4. First and last stages of a proof: A sequence containing only the first and the last reasoning stages. In this case the learner has to arrange the order of a proof simultaneously with evaluating some inference rules. The learner evaluates if the rules of inference fit in different possible ways in order to build the missing reasoning stages. This case is related to a very commonly used proving technique, where one is building a proof from bottom-conclusion to top-premises in a trial-and-error procedure.

5. Multiset of premises and open assumptions: A multiset which contains at least the premises and necessary open assumptions. This case can represent any real-life reasoning task where only premises and conclusion are given.

### 5.2.1 Stream of positive data: sequences of proofs

Since the set of propositional formulas is a countable set, we can use $\mathbb{N}$ to enumerate it. Note that $\mathcal{F}^{\leq 3}$ has more than countably many elements, to be precise it has cardinality $2^{\aleph_0}$. Our learning framework requires countable streams of data, so we need to be sure that the generated sets $\langle R \rangle$ are countable sets in order for the countable environments to be able to enumerate all true information. We also need to be sure that in every stream of data for each inference system $R_i$, every inference function in $R_i$ is used at least once.

**Lemma 1** *If $R \subseteq \mathcal{F}^{\leq 3}$ is finite, then $\langle R \rangle$ is also countable.*

*Proof:* By definition, each element of $\langle R \rangle$ is a finite sequence $\mathcal{S}_0, \ldots, \mathcal{S}_n$ of finite tuples of finite sets over a countable set of propositional formulas. That gives us that there are countably many reasoning sequences $\hat{x}$, and since in each case $y \in \mathcal{S}_n$ where $\mathcal{S}_n$ is a finite set; then there are countably many pairs $(\hat{x}, y) \in \mathcal{R}$. □

**Corollary 1** *$\langle R_i \rangle$ is countable for every $R_i \in \mathcal{R}$.*

The agent receives elements of an environment for a set of rules. The agent has to learn an inference system from proofs that correspond to the set of all proofs generated by the inference system.

**Definition 32** *Let $R \in \mathcal{R}$. A* text *$\epsilon = ((\hat{x}, y)_n)_{n \in \mathbb{N}}$ in $\langle R \rangle$ is a sequence of R-proofs from $\langle R \rangle$ enumerating all and only the elements from $\langle R \rangle$ allowing repetitions.*

To illustrate the definition above, consider the following example.

**Example 7** *Let $A, B$ be propositional formulas. Abusing from notation, we will use $A \cup B$ to denote the operation $\{A\} \cup \{B\}$ that may occur in the set of assumptions $\Gamma$. Take $(\hat{x}, y)$ such that:*

- *$\hat{x} = \mathcal{S}_0, \mathcal{S}_1$ such that;*
  - *$\mathcal{S}_0 = \{((p; \emptyset), p)_1, ((q; \emptyset), q)_2\}$,*
  - *$f_{\wedge I} : \mathcal{S}_1 = \{((p; \emptyset), p)_1, ((q; \emptyset), q)_2, ((p \cup q; \emptyset), p \wedge q)_3\}$,*
  - *$f_{\wedge I}(((p; \emptyset), p)_1, ((q; \emptyset), q)_2) = ((p \cup q; \emptyset), p \wedge q)_3$*
- *$y = ((p \cup q; \emptyset), p \wedge q)$ (which is an element of $\mathcal{S}_1$)*

*Take any $\epsilon$ text for $\langle R_{ND} \rangle$, then for some $k \in \mathbb{N}$ $\epsilon_k = (\hat{x}, y)$.*

**Definition 33** *We will use the following notation:*

1. *$\epsilon_n$ is the n-th element of $\epsilon$;*
2. *$\epsilon \upharpoonright n$ is the sequence $(\epsilon_0, \epsilon_2, \ldots, \epsilon_n)$;*
3. *$G(\{\epsilon_n\})$ will denote the set of all rules that were used in proof $\epsilon_n$. Similarly for $G(\epsilon \upharpoonright n)$, the set of all rules used in complete proofs for $\epsilon_1, \ldots, \epsilon_n$ for every $n \in \mathbb{N}$. In general for every $S \subseteq \langle R \rangle$, $G(S)$ is the set of all rules used in complete proofs of the data items appearing in $S$.*
4. *$set(\epsilon \upharpoonright n)$ is the set of elements that appear in $\epsilon \upharpoonright n$.*

Note that clearly, there will be cases in which for some different systems $R, R' \in \mathcal{R}$, finite parts of data streams will be the same. This means that on a given text learner will not be able to distinguish two systems right away as in Example 8 below. This is precisely what makes the learnability of class $\mathcal{R}$ interesting.

**Example 8** *Assume the agent needs to learn natural deduction inferential system, i.e., $R^T = R_{ND}$. So the agent will be entertained with positive data from $R_{ND}$. Suppose $R_1 := (R_{ND} \setminus \{[f^r_{\vee I}] \cup [f_{\wedge I}] \cup [f^l_{\vee I}]\}) \cup \{[f_1] \cup [f_2] \cup [f_3]\}$ such that $f_1((\Gamma, A)) = (\Gamma, A \wedge B)$, $f_2((\Gamma, A), (\Gamma, B)) = (\Gamma, A \vee B)$ and $f_3((\Gamma, B)) = (\Gamma, A \wedge B)$. Now take assumptions $\{p, q; \emptyset\}$ and conclusion $p \vee q$. Clearly we can provide a $R_{ND}$-proof by only one rule instance of $[f_{\vee I}]$ for premise $p$. Now we provide an $R_1$-proof for this case in which we specify the rule for each reasoning stage extension,*

- $\mathcal{S}_0 = \{((p \cup q; \emptyset), p)_1, ((p \cup q; \emptyset), q)_2\}$;
- $f_1^1 : \mathcal{S}_1 = \{((p \cup q; \emptyset), p)_1, ((p \cup q; \emptyset), q)_2, ((p \cup q; \emptyset), p \wedge q)_3\}$, *the agent obtained* $((p \cup q; \emptyset), p \wedge q)$ *by applying rule* $[f_1]$ *to input* $((p \cup q; \emptyset), p)_1$;
- $f_{\wedge E}^3 : \mathcal{S}_1 = \{((p \cup q; \emptyset), p)_1, ((p \cup q; \emptyset), q)_2, ((p \cup q; \emptyset), p \wedge q)_3, ((p \cup q; \emptyset), q)_4\}$, *note that here the agent obtained* $((p \cup q; \emptyset), q)$ *again by means of rule* $[f_{\wedge E}^l]$;
- $f_2^{1,4} : \mathcal{S}_1 = \{((p \cup q; \emptyset), p)_1, ((p \cup q; \emptyset), q)_2, ((p \cup q; \emptyset), p \wedge q)_3, ((p \cup q; \emptyset), q)_4, ((p \cup q; \emptyset), p \vee q)_5\}$.

*Clearly the inputs for rule* $[f_2]$ *were* $((p \cup q; \emptyset), p)_1$ *and* $((p \cup q; \emptyset), q)_4$ *in order to obtain* $((p \cup q; \emptyset), p \vee q)_5$.

## 5.3 Unsupervised learning

In this section we will explore the first three methods of information mentioned above without a teacher. We will analyze them in the decreasing order of informativeness. This apparent difficulty factor will play a significant role for learnability of the class $\mathcal{R}$.

### 5.3.1 Fully labeled proofs

This method of information corresponds to an extra-informative environment, i.e., proofs with rule and input specification. The learner acquires the inference system which contains precisely the rules that were used in $\hat{x}$ for concluding $y$. We will use symbol $\langle R \rangle^f$ to denote that the $R$-proofs in $\langle R \rangle$ have rule and input specification.

**Observation 3** *Whenever* $R_i \neq R_j$, *then also* $\langle R_i \rangle^f \neq \langle R_j \rangle^f$ *since each text* $\epsilon$ *for* $R_i$ *should contain at least one pair* $(\hat{x}, y)$ *in which* $\hat{x}$ *indicates the application for an* $[f'] \in R_i$ *which is not contained in* $R_j$ *(or vice versa).*

Since the rules are explicitly given in every data item, the learner can just accumulate the ones observed and output the set each time. Let $\mathcal{R}' \subseteq \mathcal{R}$, we will use $sm\{R \in \mathcal{R}'\}$ to denote the set with the smallest cardinality among $\mathcal{R}'$. Thus, let us define a learner in the following way:

$$L(\epsilon \restriction n) = sm\{R_i \in \mathcal{R}\} \text{ such that } G(\epsilon \restriction n) \subseteq R_i$$

The function $L$ outputs an hypothesis $R_i$ with the smallest cardinality among the possible alternatives in $\mathcal{R}$ such that it contains every rule the learner encountered in the data stream. He can change his hypothesis several times until at some data item $k$, all his future guesses will be the same.

**Proposition 4** $\mathcal{R}$ *is identified in the limit by the learner* $L$.

*Proof:* Let $\epsilon$ be any text for $R_i$. Since $\epsilon$ contains all fully labelled $R$-proofs with their respective rule instances in each step, and $R_i$ is finite, after some $m \in \mathbb{N}$ we will have that $\epsilon \restriction m$ has made use of all rules in $R_i$. Therefore $G(\epsilon \restriction m) \subseteq R_i$. For all $n \geq m$, $G(\epsilon \restriction n) \subseteq R_i$. It could be that another $R_j$ satisfies this condition, if that is the case then clearly $R_i \subseteq R_j$. Since $L$ chooses the smallest set satisfying the conditions, $L(\epsilon \restriction n) = R_i$ for all $n \geq m$. □

Observe that this method of information allows a straightforward disambiguation between two systems due to the rule specification in each item. This is precisely the reason why this method of information is *too* informative. An alternative proof for Proposition 4 can be formulated by means of Angluin's characterization theorem in terms of FTT sets. This strategy will be used in the next section.

### 5.3.2 Partially labeled proofs

This method of information is also quite informative. It is specified which inputs the rules took to extent the reasoning stages of the proof. The learner evaluates if the rules of his current hypothesis fit in the proof, i.e., if they could take the respective inputs and output the following stages. For this case we will use $\langle R \rangle^{inp}$ to denote that the inputs are being specified at each stage in every $R$-proof from $\langle R \rangle$.

**Observation 4** *Whenever $R_i \neq R_j$, then also $\langle R_i\rangle^{inp} \neq \langle R_j\rangle^{inp}$ since in each text $\epsilon$ for $R_i$ there is at least one pair $(\hat{x}, y)$ in which $\hat{x}$ contains two consecutive stages $\mathcal{S}_i, \mathcal{S}_{i+1}$ such that $[f] \in R_i$ (which is not contained in $R_j$) takes precisely the indicated inputs in $\mathcal{S}_i$ and outputs the new element which appears in $\mathcal{S}_{n+1}$. To see this, it is enough to observe that all rules in $\bigcup \mathbb{B}$ differ with respect to their input-output behaviour, i.e., they differ on the arity or on the shape of the output formula.*

**Proposition 5** *For each $R_i \in \mathcal{R}$, $\langle R_i\rangle^{inp}$ is a recursive set.*

*Proof*: For every $R_i \in \mathcal{R}$ we define the recursive function,

$$g_i((\hat{x}, y)) = 1 \text{ iff } \hat{x} \text{ is a proof of } y \text{ in which at each stage } k \text{ in } \hat{x}, \text{ there is a rule } [f_r] \in R_i \text{ that precisely with the indicated inputs, outputs what extends stage } k \text{ into stage } k+1, \textit{i.e.}, \text{ there is } R' \subseteq R_i \text{ such that } [f_r] \in R' \text{ was used in } \hat{x}.$$

Note that each step in the proof had been indicated in $\hat{x}$. Thus, it is easy to see that $g_i$ is recursive since it only needs to perform a finite verification of the rules in $R_i$, i.e., if some rules in $R_i$ can be accommodated with the indicated inputs in the finite sequence of reasoning stages appearing in $\hat{x}$ that has $y$ as the desired conclusion. □

The result in Proposition 5 leads us to the following observation about the learning space.

**Corollary 2** *The class $\mathcal{R}$ of alternative inference systems is decidable.*

*Proof:* We define $h$ to be the following decision function,

$$h((\hat{x}, y), i) = \begin{cases} 1 & \text{if } g_i((\hat{x}, y)) = 1 \\ \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

which is recursive since $g_i$ is recursive for each $i \in I_{\mathcal{R}}$.□

In what follows, we will see two alternative approaches for proving learnability of the class $\mathcal{R}$. The first proof will be by means of Angluin's characterization theorem. The second one, will be by using the result obtained in Corolary 2.

Recall Example 7 where the complete proof in $\hat{x}$ by using $R_{ND}$ consists of only two reasoning stages, i.e., where only one rule application of rule $[f_{\wedge I}]$ was needed. Observe that there will be similar simple data items (with two reasoning stages and atomic propositional formulas) for each rule in $R_{ND}$; and these data items will be spread among any text of $R_{ND}$. As a matter of fact, this will be the case for any alternative system $R \in \mathcal{R}$. We will see the importance of this observation in the following result.

**Theorem 2** *$\mathcal{R}$ is identifiable in the limit under $\langle R\rangle^{inp}$ method of information.*

*Proof:* It suffices to show that there is a FTT subset $D_i$ of $\langle R_i\rangle^{inp}$ for each $R_i \in \mathcal{R}$. First we define the set $X^i_{[f_k]}$,

$$(\hat{x}, y) \in X^i_{[f_k]} \text{ only if } \hat{x} \text{ is a one-step proof with only an instance of } [f_k] \in R_i.$$

So $X^i_{[f_k]}$ is the set containing all proofs with only two reasoning stages using rule $[f_k] \in R_i$, i.e., $P = \{\mathcal{S}_0, \mathcal{S}_1\}$ such that $\mathcal{S}_1$ was obtained by applying only once rule $[f_k] \in R_i$ and $y \in \mathcal{S}_1$. Observe that the $X^i_{[f_k]}$ are infinite. We will use these sets to build the FTT's. Note that we have one $X^i_{[f_k]}$ for each rule in $R_i$. Now, from each $X^i_{[f_k]}$, we select a "minimal witness" of the set. So a FTT for $R_i$ (called $D_i$) is the set of these minimal witnesses taken from each $X^i_{[f_k]}$. By *minimal* we mean the first element occurring in $X^i_{[f_k]}$ with atomic propositional formulas. Formally: take a minimal element $(\hat{x}, y)_k$ of $X^i_{[f_k]}$ for each rule $[f_k] \in R_i$, then a FTT set $D_i$ for $R_i$ will be the union of these elements $(\hat{x}, y)_k$. Note that $D_i$ is a copy of $R_i$ but in terms of the sequences of length two obtained by a single application of its rules. Lets verify that $D_i$ is indeed a FTT set for $R_i$. Clearly since $R_i$ is finite, $D_i$ is finite. What remains to be proven is the following: If $D_i \subseteq \langle R_j\rangle^{inp}$, then $\langle R_j\rangle^{inp} \not\subset \langle R_i\rangle^{inp}$. Towards contradiction suppose $\langle R_j\rangle^{inp} \subset \langle R_i\rangle^{inp}$. There is $[f_r] \in R_i$ such that $[f_r] \notin R_j$. Therefore $\langle R_j\rangle^{inp}$ does not contain

proofs which makes use of rule $[f_r]$, in particular single instances of $[f_r]$. By definition $D_i$ contains an $R$-proof which is obtained by a single instance of $[f_r]$ which contradicts the fact that $D_i \subseteq \langle R_j \rangle^{inp}$. □

Using function $h$, we can define a learning function that identifies $\mathcal{R}$ in the limit.

$$L(\epsilon \restriction n) = sm\{R_i \in \mathcal{R}\} \text{ such that } \forall k \leq n,\ h((\hat{x}, y)i) = 1$$

Our learning function $L$ outputs an hypothesis $R_i$ with the smallest cardinality from the possible alternatives in $\mathcal{R}$ when it is sure that every proof data item $\epsilon_k$ presented in $\epsilon \restriction n$ is an $R_i$-proof. Learner $L$ is able to disambiguate $R^T$ from the rest $R_j \in \mathcal{R}$ when simple proofs of rule instances are encountered in $\epsilon$.

**Proposition 6** *$\mathcal{R}$ is identifiable in the limit by learner $L$.*

*Proof:* Let $\epsilon$ be any text for $R_i$. Since $\epsilon$ contains all stage-by-stage proofs and $R_i$ is finite; after some $m \in \mathbb{N}$ we will have that: all steps in each proof in $\epsilon \restriction m$ were governed by a rule from some $R' \subseteq R_i$ with the indicated inputs; and all $[f_r] \in R_i$ will be used at least once in $\epsilon \restriction m$. For all $n \geq m$, we will find $R' \subseteq R_i$ such that at each stage $\mathcal{S}$ of a proof $\hat{x}$ of any data item $(\hat{x}, y)$ there is a rule $[f_r] \in R'$ which was used to extend a previous stage to obtain stage $\mathcal{S}$. Thus $h((\hat{x}, y)i) = 1$. It could be that another $R_j$ satisfies this conditions, if that is the case then $R_i \subseteq R_j$. Since $L$ chooses the smallest set satisfying the conditions, $L(\epsilon \restriction n) = R_i$ for all $n \geq m$. □

Observe that since our function $h$ is computable, our learner $L$ is computable too. Observe also that the procedure we described for obtaining the FTT's is computable. An alternative way to prove Proposition 6 is by means of the FTT's defined for Theorem 2. However, this method of information is informative enough in order for the learner to disambiguate between two or more alternatives without using the FTT sets.

### 5.3.3 Non-labeled proof sequence

This method of information corresponds also to an informative environment. Now however it does not explicitly say about the rules. We could think again of a classroom scenario, now one in which the learner is guessing which rules where used in each step of a proof and which inputs should be considered in each reasoning stage for extending it. The learner evaluates if the rules of $R$ fit in the proof steps. In this case the pieces of data come simply from the set $\langle R \rangle$.

**Observation 5** *Whenever $R_i \neq R_j$, then also $\langle R_i \rangle \neq \langle R_j \rangle$ since in each stream of data $\epsilon$ for $R_i$ should be at least one pair $(\hat{x}, y)$ in which $\hat{x}$ contains a single instance of an $[f] \in R_i$ which is not contained in $R_j$ (or vice versa); and the assumptions (premises and open assumptions) in the arguments of the rule at each stage in $\hat{x}$ are precisely the necessary (i.e., all and only) the assumptions required for the rule to go through. Inference rules are characterized not only by their behaviour with the relation between the formulas that appear in the inputs and output, but also by their treatment to specifically the* necessary *assumptions in the input-output relation.*

To illustrate Observation 5, we provide the following example of data items $(\hat{x}, y)_1$ and $(\hat{x}, y)_2$. They are similar, but $(\hat{x}, y)_1$ contains redundant assumptions. Their presence makes it impossible to decide which rule has been implemented to transition from $\mathcal{S}_0$ to $\mathcal{S}_1$. On the other hand, the minimal set of necessary assumptions as in $(\hat{x}, y)_2$ allows such disambiguation.

**Example 9** *Consider the data item $(\hat{x}, y)_1$ such that $\hat{x} = \{\mathcal{S}_0, \mathcal{S}_1\}$,*

$$\mathcal{S}_0 = \{((\emptyset; \{p\} \cup \{q\}), p), ((\emptyset; \{p\} \cup \{q\}), q)\}$$

*and*

$$\mathcal{S}_1 = \{((\emptyset; \{p\} \cup \{q\}), p), ((\emptyset; \{p\} \cup \{q\}), q), ((\emptyset; \{p\} \cup \{q\}), p \wedge q)\}.$$

*Let $[f_r]$ be the rule which takes an input of the form $(\Gamma, A)$ and outputs $(\Gamma, A \wedge B)$. Then, we will not be able to disambiguate between $[f_r]$ and $[f_{\wedge I}]$ for the rule used to extend $\mathcal{S}_0$. Note that $[f_r]$ will take input $((\emptyset; \{p\} \cup \{q\}), p)$ and $[f_{\wedge I}]$ will take inputs $((\emptyset; \{p\} \cup \{q\}), p)$ and $((\emptyset; \{p\} \cup \{q\})q)$; and both of them*

*output* $((\emptyset; \{p\} \cup \{q\}), p \wedge q)$.

*However if* $(\hat{x}, y)_2$ *is such that* $\hat{x} = \{\mathcal{S}_0, \mathcal{S}_1\}$ *where*

$$\mathcal{S}_0 = \{((\emptyset; \{p\}), p), ((\emptyset; \{q\}), q)\}$$

*and*

$$\mathcal{S}_1 = \{((\emptyset; \{p\}), p), ((\emptyset; \{q\}), q), ((\emptyset; \{p\} \cup \{q\}), p \wedge q)\}.$$

*we will be able to disambiguate between the two, since we focus also on the relation between the assumptions of the input/inputs and the assumptions of the output; and in this case the relation corresponds to* $[f_{\wedge I}]$. *But observe that to* localize *this relation we required to only have the necessary assumptions.*

Since our texts are complete, i.e., they contain all proofs, such minimal items in Observation 5 will appear.

The constraints for the usage of a rule depend only on the treatment of the necessary assumptions (premises and open assumptions). Rules do not say anything about premises or open assumptions that are not being considered for the rule application (even though they can be *present*). That said, *rule usage* seems to be characterized by the treatment of the necessary assumptions under consideration.

**Proposition 7** *For each* $R_i \in \mathcal{R}$, $\langle R_i \rangle$ *is a recursive set.*

*Proof*: For every $R_i \in \mathcal{R}$ we define the recursive function,

$g_i((\hat{x}, y)) = 1$ iff $\hat{x}$ is a proof of $y$ in which at each stage $k$ in $\hat{x}$, there is a rule $[f_r] \in R_i$ that takes inputs at stage $k$ which outputs precisely what extends stage $k$ into stage $k+1$, *i.e.* there is $R' \subseteq R_i$ such that $[f_r] \in R'$ was used in $\hat{x}$.

Note that each step in the proof had been indicated in $\hat{x}$. Thus, $g_i$ is recursive since it only needs to perform a finite number of trials for verification between the rules in $R_i$ and a finite number of possible inputs in a finite number of reasoning stages, i.e., if some rules can be accommodated with some inputs in the finite sequence of reasoning stages appearing in $\hat{x}$ that has $y$ as the desired conclusion. □

Note that here we obtain a decision function $h$ as the one defined for partially labelled proofs for class $\mathcal{R}$.

In what follows, we will again addressed two alternative approaches to *attempt* a proof for learnability of the class $\mathcal{R}$. The first sketch of a proof will be by means of Angluin's characterization theorem, we do not claim that such sketch proof really covers all details (a complete proof concerning rule-usage-characterization in terms of necessary assumptions remains necessary for finding such FTT's).
Then we will use $g_i$ functions defined in 7 and FTT sets. This time the learner also needs to perform a search for appropriate item inputs using the necessary assumptions. Moreover, the FTT sets are a bit more restricted this time, now we also require that the elements contain only necessary assumptions.

**Conjecture 1** $\mathcal{R}$ *is identifiable in the limit when the method of information is* $\langle R \rangle$.

*Proof sketch:* It suffices to be shown that there is a FTT subset $D_i$ of $\langle R_i \rangle$ for each $R_i \in \mathcal{R}$. We define,

$$(\hat{x}, y) \in X^i_{[f_k]} \text{ only if } \hat{x} \text{ is a proof that requires only one rule instance of } [f_k] \in R_i.$$

So $X^i_{[f_k]}$ is the set containing all proofs with only two reasoning stages using rule $[f_k] \in R_i$, i.e., $P = \{\mathcal{S}_0, \mathcal{S}_1\}$ such that $\mathcal{S}_1$ was obtained by applying only once rule $[f_k] \in R_i$ and $y \in \mathcal{S}_1$. Take the minimal element $(\hat{x}, y)_k$ of $X^i_{[f_k]}$ (by *minimal* we mean the first element occurring in $X^i_{[f_k]}$ with atomic propositional formulas ) for each rule $[f_k] \in R_i$ such that the assumptions accompanying the formulas in the arguments appearing in $\hat{x}$ and $y$ are exactly the necessary ones. Then a FTT set $D_i$ for $R_i$ will be the union of these minimal elements $(\hat{x}, y)_k$ with necessary assumptions. Note that $D_i$ is a copy of $R_i$ but in terms of single proofs requiring rule instances from rules in $R_i$. Lets verify that it is indeed a DFTT set for $R_i$. Clearly since $R_i$ is finite, $D_i$ is finite. What remains to be proved is the following: If $D_i \subseteq \langle R_j \rangle$, then $\langle R_j \rangle \not\subset \langle R_i \rangle$. By contradiction suppose $\langle R_j \rangle \subset \langle R_i \rangle$. There is $[f_r] \in R_i$ such that $[f_r] \notin R_j$. Therefore $\langle R_j \rangle$ does not contain proofs which makes use of rule $[f_r]$, in particular single instances of $[f_r]$. By definition $D_i$ contains a proof which requires a single instance of $[f_r]$ with the corresponding necessary assumptions which contradicts the fact that $D_i \subseteq \langle R_j \rangle$. □

Using functions $g_i$, we can define a learning function that identifies $\mathcal{R}$ in the limit. This time our learning function will need to really pay attention to the $FTT$'s considering the necessary assumptions presented in the data items. Otherwise disambiguation between two or several alternatives may not be evident. Recall that in previous cases, there was no need for the learner to *look for* FTT's since disambiguation came quickly enough in the process and by other means.

$$L(\epsilon \restriction n) = sm\{R_i \in \mathcal{R}\} \text{ such that } \forall k \leq n,\, g_i(\epsilon_k) = 1 \text{ and } D_i \subseteq set(\epsilon \restriction n)$$

Our learning function $L$ outputs an hypothesis $R_i$ with the smallest cardinality from the possible alternatives in $\mathcal{R}$ that she is sure every proof data item $\epsilon_k$ presented in $\epsilon \restriction n$ is an $R_i$-proof. Learner $L$ is able to disambiguate $R^T$ from the rest $R_j \in \mathcal{R}$ when simple proofs of rule instances and only the necessary assumptions are encountered in $\epsilon$.

**Conjecture 2** *$\mathcal{R}$ is identifiable in the limit via learner $L$.*

*Proof sketch:* Let $\epsilon$ be any text for $R_i$. Since $\epsilon$ contains all stage-by-stage proofs and $R_i$ is finite; after some $m \in \mathbb{N}$ we will have that: all steps in each proof in $\epsilon \restriction m$ were governed by a rule from some $R' \subseteq R_i$ with precisely the relation between assumptions that needed to be verified; and all $[f_r] \in R_i$ will be used at least once in $\epsilon \restriction m$. For all $n \geq m$, we will find $R' \subseteq R_i$ such that at each stage $\mathcal{S}$ of a proof $\hat{x}$ of any data item $(\hat{x}, y)$ there is a rule $[f_r] \in R'$ which was used to extend a previous stage to obtain stage $\mathcal{S}$. Thus, $g_i(\epsilon_k) = 1$ and $D_i \subseteq \epsilon \restriction n$. It could be that another $R_j$ satisfies this conditions, if that is the case then $R_i \subseteq R_j$. Since $L$ chooses the smallest set satisfying the conditions, $L(\epsilon \restriction n) = R_i$ for all $n \geq m$. □

### 5.3.4 Less informative data

In this section we will explore the last two methods of information mentioned at the beginning of this chapter. First by means of unsupervised learning and then by implementing a teacher. We will first address method 4 i.e., the first and last stages in a proof; then we will address method 5, i.e., set of premises and open assumptions. The difficulty factor between these two and the three previously discussed methods will play a significant role for learnability of the class $\mathcal{R}$.

**The first and last stages in a proof** This method of information corresponds to a weaker informative environment than the previous cases. In this method, all the conclusions of the stages at each step in the proof are given but the right order of the proof stages is unknown. Note that the last stage $\mathcal{S}_n$ contains all the conclusions obtained in the reasoning process. So it contains all the inputs and conclusions from the previous stages, the proof needs to be "put in order" by means on extending correctly the first stage $\mathcal{S}_0$ with the elements in $\mathcal{S}_n \setminus \mathcal{S}_0$ and so on. This is a more complicated task but it can be done *backwards*, as Sherlock Holmes would say, a very useful practice:

> *I'm solving a problem of this sort, the grand thing is to be able to reason backwards. That is a very useful accomplishment, and a very easy one, but people do not practice it much. In the every day affairs of life it is more useful to reason forwards, and so the other comes to be neglected. There are fifty who can reason synthetically for one who can reason analytically...*

The learner has to arrange the order of a proof at the same time that he is evaluating the rules in $R$. The learner evaluates if the rules of $R$ fit in different possible ways of filling the missing stages in $\hat{x}$. Note that there is more than one way of ordering a set of deductive stages in order to obtain a successful proof, but there are finitely many. This feature increases the level of complexity of the learning task.

We will use $\langle R \rangle^{set}$ to denote that from every $R$-proof in $\langle R \rangle$ we are only presenting the first and last stages.

**Definition 34** *Let $P_{\Gamma,y} := \mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_n$ be any $R$-proof from $\Gamma$ to $y$. We will use $\mathcal{S}_{initial}$ to denote $\mathcal{S}_0$ and $\mathcal{S}_{final}$ to denote $\mathcal{S}_n$.*

**Observation 6** *Whenever $R_i \neq R_j$, then also $\langle R_i \rangle^{set} \neq \langle R_j \rangle^{set}$. This is because each stream of data $\epsilon$ for $R_i$ should contain at least one pair $(\hat{x}, y)$ containing* only necessary *assumptions in the arguments and in which $\hat{x}$ requires from stage $\mathcal{S}_{initial}$ to stage $\mathcal{S}_{final}$ a single instance of precisely the rule $[f] \in R_i$ which is not contained in $R_j$ (or vice versa). This refers to data items in which when $\mathcal{S}_{initial} = \mathcal{S}_0$, it happens to be the case that $\mathcal{S}_1 = \mathcal{S}_{final}$.*

**Proposition 8** *For every $R_i \in \mathcal{R}$, $\langle R_i \rangle$ is a recursive set.*

*Proof:* For every $R_i \in \mathcal{R}$ we define a recursive function $g_i$ as,

$$g_i((\hat{x}, y)) = 1$$
iff
there is an order $\dot{x}$ of $\mathcal{S}_{final} \setminus \mathcal{S}_{initial}$
such that $\dot{x}$ is used to form the missing stages from $\mathcal{S}_{initial}$ to obtain a proof of $y$; and there is a rule $[f_r] \in R_i$ that outputs precisely what extends a stage , *i.e.* there is $R' \subseteq R_i$ such that $[f_r] \in R'$ was used in building missing stages using $\dot{x}$.

Note that each step in the proof had been indicated in $\hat{x}$. Thus, $g_i$ is recursive since it only needs to perform a finite number of order trials for verification attempts between the rules in $R_i$ and a finite number of possible inputs in a finite number of reasoning stages, i.e., if for some order some rules can be accommodated with some inputs in the finite sequence of reasoning stages appearing in $\hat{x}$ that has $y$ as the desired conclusion. □

As in previous cases, we obtain a decision function $h$ for the class $\mathcal{R}$.

The information of the propositions leading to $y$ is in $\hat{x}$ but not the structure itself. As we mentioned before, a possible strategy for the learner is to start with conclusion $y$ constructing the proof from bottom to top as it is used in real practice. We conjecture that considering data items containing *precisely necessary* assumptions will play a sufficient role for disambiguation, thus the issue of learnability of the class $\mathcal{R}$ under this method of information is similar to the case of non-labelled proofs.

**Set of premises and necessary open assumptions** This method of information can represent daily life conversations between people or reasoning tasks in which people need to accommodate the right open assumptions (if any) with the given premises while reasoning. In other words they need to simultaneously evaluate premises and possible open assumptions while trying to construct a proof. Imagine a scenario where a detective is solving a case, or a debate is taking place. There are many real-life reasoning environments that adopt this form of presenting information.

This case is more complex than the previous ones because a complete reconstruction of the proof sequence needs to be done. We motivate this claim with a quote from Sherlock Holmes book series,

> *Most people, if you describe a train of events to them, will tell you what the result would be. They can put those events together in their minds, and argue from them that something will come to pass. There are few people, however, who, if you told them a result, would be able to evolve from their own inner consciousness what the steps were which led up to that result. This power is what I mean when I talk of reasoning analytically.*

Several types of confusions can evolve besides the confusion of an inference system. Confusion can also arise from the wrong usage of premises and open assumptions in any reasoning process. Confusions of this type and confusion on the inference system can occur simultaneously. It could be that the learner is in a wrong inference system and by misusing the given open assumptions in $\hat{x}$ he obtains the desired conclusion $y$. It can be the case that he continues misusing open assumptions obtaining "proofs" which make him believe that the misleading inference system is the correct one. It seems that in order to prevent this we need to put some constraints on the usage of open assumptions.

We could ask for the learner to have basic background knowledge on how to use open assumptions. Therefore, we assume the learner knows the appropriate usage of open assumptions according to the rules in his current system. For instance when $R_{ND}$ is his current system, the learner knows the following:

1. The learner knows that open assumptions are sub-formulas of the formulas occurring either in premises or in conclusion.

2. The learner knows that he cannot conclude $y$ if:

   - there are still open assumptions that need to be dropped;
   - $y$ was obtained from only one of the disjoints of a given disjunction.

These constraints could be lifted in order to analyze simultaneously the process of learning the usage of open assumptions. We leave this topic to further research.

We will use $\langle R\rangle^{p\&a}$ to denote that from the $R$-proofs in $\langle R\rangle$ we are only presenting the premises and open assumptions needed for a proof.

In this method of information, we will clearly have cases in which $R_i \neq R_j$ but a pair $(\hat{x}, y)$ can be proved in both systems by different rules. To illustrate this phenomena consider the following example.

**Example 10** *Assume the agent needs to learn natural deduction inferential system, i.e., $R^T = R_{ND}$. So the agent will be entertained with positive data from $R_{ND}$. Suppose $R_1 := (R_{ND} \setminus \{[f^r_{\vee I}] \cup [f_{\wedge I}] \cup [f^l_{\vee I}]\}) \cup \{[f_1] \cup [f_2] \cup [f_3]\}$ such that $f_1((\Gamma, A)) = (\Gamma, A \wedge B)$, $f_2((\Gamma, A), (\Gamma, B)) = (\Gamma, A \vee B)$ and $f_3((\Gamma, B)) = (\Gamma, A \wedge B)$. Note that this is a strange case, since suggests that the agent is confusing $\wedge$ with $\vee$; but only for the introduction rules not for the elimination rules. Suppose that for some $\epsilon_k$ all $R \in \mathcal{R}$ which are different from $R_{ND}$ and $R_1$ got dismissed. So the only remaining inference systems are these two. Now suppose $\epsilon_{k+1}$ is the pair $(\hat{x} := \{(p \wedge q) \to r, q \to p, q\}; y := r)$ in which the elements in $\hat{x}$ are premises. It is easy to see that both $R_{ND}$ and $R_1$ can provide proofs for such a pair with their respective inference rules.*

As a matter of fact it seems that their streams of data "share" a lot of items (if not all). Another example easy to verify in which both systems can provide respective proofs is $(\hat{x} := \{p \to q, q \to r\}; y := p \to (q \wedge r))$ where the elements in $\hat{x}$ are premises. Both systems can provide respective proofs for simple items, consider the following explicit example.

**Example 11** *Consider again data item $(\hat{x} := \{p, q\}; y := p \vee q)$. Clearly we can provide a $R_{ND}$-proof by only one rule instance of $[f_{\vee I}]$ for premise $p$. Now we provide an $R_1$-proof for this item with rule specification,*

- $\mathcal{S}_0 = \{((p \cup q; \emptyset), p)\}$;
- $f_1 : \mathcal{S}_1 = \{((p \cup q; \emptyset), p), ((p \cup q; \emptyset), p \wedge q)\}$, *the agent obtained $((p \cup q; \emptyset), p \wedge q)$ by applying rule $[f_1]$;*
- $f^l_{\wedge E} : \mathcal{S}_1 = \{((p \cup q; \emptyset), p), ((p \cup q; \emptyset), p \wedge q), ((p \cup q; \emptyset), q)\}$, *note that here the agent obtained $((p \cup q; \emptyset), q)$ by means of rule $[f^l_{\wedge E}]$;*
- $f_2 : \mathcal{S}_1 = \{((p \cup q; \emptyset), p), ((p \cup q; \emptyset), p \wedge q), ((p \cup q; \emptyset), q), ((p \cup q; \emptyset), p \vee q)\}$.

*Clearly the inputs for rule $[f_2]$ were $((p \cup q; \emptyset), p)$ and $((p \cup q; \emptyset), q)$ in order to obtain $((p \cup q; \emptyset), p \vee q)$.*

*This is independent from our proof representation. Observe that we can construct a step-by-step linear representation using the rules that correspond to the rules in $R_1$,*

| | | |
|---|---|---|
| 1. | $p$ | premise |
| 2. | $p \wedge q$ | rule $[f_1]$ applied to 1 |
| 3. | $q$ | rule $[f^l_{\wedge E}]$ applied to 2 |
| 4. | $p \vee q$ | rule $[f_2]$ applied to 1 and 3 |

How can we make the learner $L$ disambiguate between $R_{ND}$ and $R_1$ ? We would like the learner to "realize" his misinterpretation at some point. In the previous example the learner will " acknowledge" his mistake when he encounters $\epsilon_m$ which cannot be proved by $R_1$. But it might take a lot of data items and possibly very complicated ones for him to realize his mistake. Is it possible for the learner to disambiguate between any pair $R_i, R_j$ of inference systems?

Now the question is: Can it be that $R_i \neq R_j$ but $\langle R_i\rangle^{p\&a} = \langle R_j\rangle^{p\&a}$? We believe that the answer to this question is negative. Since different systems should prove different formulas (specially since we do not have any equivalence relation in $FORM$). However, it could be that from only positive data is very hard to disambiguate between two or more possible alternatives (as in examples 10 and 11). An interesting case comes with the following example.

**Example 12** *Consider the inference systems $R_{ND}$ and $R_2$ such that*

$$R_2 = (R_{ND} \setminus \{[f_{\to E}], [f_{\to I}]\}) \cup \{[f_1], [f_2]\}$$

*such that*

- $f_1((\Gamma, \phi \to \psi); (\Gamma, \psi)) = (\Gamma, \phi)$ *which comes from* $\mathcal{B}_{\wedge I}$ *since* $f_1((\Gamma, A); (\Gamma, B)) = D$ *where* $D$ *is a subformula of* $A$*,* $A$ *is of the form* $D \to E$ *and* $B$ *is of the form* $E$*.*
- $f_2((\Gamma \cup \{A\}, B)) = (\Gamma, B \to A)$ *which comes from* $\mathcal{B}_{\to I}$

*Clearly $R_{ND}$ and $R_2$ do not have the same classes of functions, i.e., they don't have the same rules. However they prove very similar formulas ( $R_{ND}$ proves $A \to B$ only if $R_2$ proves $B \to A$ ). This means that the learner might be confused, not being able to disambiguate $R_{ND}$ from $R_2$ by means of positive data only even though they have different interpretations of $\to$.*

It should be the case that at some point the learner will be able to disambiguate between these two systems since $\langle R_{ND}\rangle^{p\&a} \neq \langle R_2\rangle^{p\&a}$, but it might take him too much time.

In many psychological experiments, this is being a recurrent mistake among participants. For instance in the Wason selection task, when participants were given the factual premise " If A then B" they interpreted that when $B$ is the case, it should be because also $A$ is the case. It means that when they were asked if they could conclude something from premises *If $A$ then $B$* and *$B$ is the case*, some participants said that they could conclude that *$A$ is the case*.

This example indicates that it is difficult to identify $\mathcal{R}$ in the limit with positive data when $\langle R\rangle^{p\&a}$ is the method of information.

What if we present a mixed stream of data to the learner? with a teacher that *supervises* the learning process? In the next chapter we will treat this problem by introducing an active teacher via negative data and counterexamples in the stream of data for learning the class $\mathcal{R}$.

## 5.4 Supervised learning: the teacher intervention

There is a lot of empirical evidence that indicates that students receiving mathematical education have difficulties with understanding the concept and process of a *mathematical proof* (55, 56 - Buchbinder and Zaslavsky, Buchbinder and Zaslavsky). On the other hand, learning theorists argue that feedback is of great importance as a means of re-evaluating false conjectures, and the use of counterexamples may help in developing more cautious concepts, in our case a more cautious use of rules of inference.[1] The studies developed by Buchbinder suggest that there are ways to incorporate, in a scholar environment, activities that could improve learning situations in which students develop their understanding of mathematical concepts and improve their reasoning skills through dealing with counterexamples. However, to really improve the efficiency of the learning process, the counterexamples need to be carefully selected by the teacher.

To illustrate the importance of *relevant* counterexamples, let us consider a doctor expert in multiple sclerosis diagnosis, who attempts to communicate the method she uses in that domain to communicate to a group of other experts. Specific positive and negative examples will form an important component of the communication. She would surely give advice about general rules, explanations of significant and irrelevant features, justifications of lines of reasoning, clarifications of exceptions. In addition, specific positive and negative examples will form an important component of the communication. As a matter of fact, the examples given are likely to be chosen so that they are *central* or *crucial* rather than random or arbitrary, in an attempt to improve understanding.

It has been shown that there are classes of grammars that cannot be learned from positive data only (5, 57). As a matter of fact, Gold's results have been taken to mean that identifying languages from positive data is too hard.

[1] Empirical studies showed that students sometimes possess wrong conceptions associated with counterexamples, their generation and usage. For instance, many students do not find a single counterexample as sufficient proof of a fallaciousness of a mathematical statement and tend to reject counterexamples or treat them as exceptions (56).

*Those working in the field generally agree that most children are rarely informed when they make grammatical errors, and those that are informed take little heed [...]*

*However, the results presented in the last section show that only the most trivial class of languages considered is learnable [...]* (5 - Gold)

In relation to that, the role of a teacher has been addressed and explored in learning theory. For instance, when learning regular languages, it can be assumed that the language is presented by an adequate Teacher, who can answer membership queries about the language, can test conjectures and can indicate whether they are equivalent to the grammar being learned, and provide counterexamples (17).

In this section we introduce the basic framework for a computational model with a learner and a teacher who interact in a sequence of episodes based on the framework presented in (57 - Angluin and Becerra-Bonache). In particular, the learner will finitely identify any inference system from class $\mathcal{R}$, when the method of information is "usually" by presenting the data as $(\epsilon_k, n)$ where $\epsilon_k = (\hat{x}, y)_k$ is in $\epsilon$ such that $\hat{x}$ contains only premises and open assumptions; $\epsilon$ will contain positive and negative items; and $n$ is a natural number which corresponds to a bound for the possible number of reasoning stages a proof for such item can contain. Additionally, we will consider a teacher who works as an oracle that helps the learner to disambiguate between elements in the class by means of presenting counterexamples when needed.

**The Learner** Our learner will be a function $L$ that attempts to construct complete proofs given arguments. The learner is expected to output a hypothesis after an initial segment of arguments of $\epsilon$. The learner starts with an arbitrary set of rules $R \in \mathcal{R}$ as his current set of rules, denoted by $R^c$. The goal of the learner is to converge to the target set of rules $R^T$ after some finite amount of data that is presented to him. He will be able to make a guess after each data item is presented. The learner does have complete knowledge about which are the alternative systems in the learning space.

**The Teacher** We represent the competence of the teacher by a finite state transducer that recognizes the target system $R_T$, and the corresponding generated set of valid proofs $\langle R \rangle$ for each element in the class. So that for every data item, he identifies the inference systems which can construct a proof item. Besides, the teacher has full knowledge of the learning space using the inference systems to identify the learner's guess. Thus he knows the order as well as for which $i \in I_{\mathcal{R}}$, $R_i$ corresponds to the learner's guess.

The teacher is assumed to answer correctly to the learner's conjectures. The answer from the teacher is *yes* or *no* depending on whether the target system $R^T$ corresponds to the conjecture. The *no* answer will be by presenting to the learner items of the form $(\epsilon_k, m, 1/0)$ for some $m \in \mathbb{N}$. Let us describe more carefully these negative answers from the teacher. Suppose that the conjecture $H := R_c$ presented by the learner does not correspond to the target system. Since the teacher has full knowledge of the learning space and all its elements, he can evaluate if either $H \subseteq R^T$, $H \supseteq R^T$ or neither of them is the case.

- If $H \subseteq R^T$ holds, then the teacher provides a counterexample for $H$ of the form $(\epsilon_k, n, 1)$. The third entry indicates to the learner that the data item $\epsilon_k$ can be proved by $R^T$ but it cannot be proved by $H$. As we mentioned before, the second entry is a bound for the *length* of the proof, in this case corresponds to the minimal number of reasoning stages in which a proof for $\epsilon_k$ by $R^T$ can be constructed. Then the learner will know that he only needs to verify the proofs of his conjecture with $n$ reasoning stages.

- If $H \supseteq R^T$ holds, then the teacher provides a counterexample for $H$ of the form $(\epsilon_k, n, 0)$. The third entry indicates to the learner that the data item $\epsilon_k$ cannot be proved by $R^T$ but it can be proved by $H$. The second entry which is the bound, corresponds to the maximal number of reasoning stages in which a proof (without redundant stages) for $\epsilon_k$ by elements in $\mathcal{R}$ can be constructed. Then the learner will evaluate if some of the alternative inference systems contain proofs up to length $n$ of this $\epsilon_k$.

- If none of the previous cases hold, the teacher provides a counterexample as in the case of $H \supseteq R^T$.

**Interaction between learner and teacher** The teacher and learner know the task, so the learner knows that he needs to converge to the target proof system. The learner receives a data item $(\epsilon_k, n)$ from a stream of data $\epsilon$ and starts with an arbitrary hypothesis $H$ from the learning space. The learner will attempt to satisfy the proof with his current set of rules restricting his proof attempts with the number of steps indicated by the learner. Only in this way he can test if his current inference system is adequate. If he manages to build a proof, he makes a conjecture to the teacher. If the teacher's answer is *yes*, he halts. If the answer is *no* then he receives a counterexample in one of the two forms described above. By means of the negative data, i.e., $(\epsilon_k, n, 0)$, the learner can determine if he is using the wrong inference system. Moreover, the learner can dismiss all the supersets of his current rejected conjecture right away, since the same counterexample serves for those alternatives. Analogously, the positive-counterexamples $(\epsilon_k, n, 1)$, i.e., the ones which are provable by the target system, will make the learner to dismiss all possible subsets of his current rejected conjecture. Therefore the learner can select from the remaining alternatives the one with the minimal index to be his new current hypothesis $R_c$. Then the learner uses this counterexample of the teacher ($(\epsilon_k, n, 0)$ or $(\epsilon_k, n, 1)$) to test the system he just chose. Thus,

- If $(\epsilon_k, n, 0)$ is the case and the learner manages to find a proof in less or equal than $n$ reasoning stages by using $R_c$, the learner dismisses such hypothesis and selects a new one. If he does not find a proof by means of at most $n$ stages, he makes a new conjecture $H := R_c$ and the teacher makes an evaluation again.
- If $(\epsilon_k, n, 1)$ is the case and the learner does not manage to find a proof in exactly $n$ reasoning stages by using $R_c$, the learner dismisses such hypothesis and selects a new one. If he does find a proof with exactly $n$ reasoning stages, the learner makes a new conjecture $H := R$ and the teacher makes an evaluation again.

The procedure presented above is depicted in Figure 5.2.

**Discussion about the teacher's adequacy** Our setting suggests that for finite identification we need an adequate teacher, and for such adequacy we are asking for a significant mathematical competence. Would a less competent teacher be sufficient?

Many of the learning models envision a teacher who interacts in some way with the learner, but basically in almost all of these models the learning process is the responsibility of the learner alone. However real-world learning is often highly teacher-dependent, thus several researchers have suggested moving some of the computation from the learner to the teacher. (58 - Goldman and Mathias) ask the question: What kind of teacher would be so smart that any reasonable student would understand the material at the end of the lecture? They introduce the notion of *teaching dimension*, which corresponds in the analogy with a classroom scenario to the length of the shortest lecture a teacher can give that will get every reasonable student to understand the concept. In the Goldman-Kearns teaching model, the teaching dimension of a learning class is the minimum number $m$ such that for every element in the class, there exists a set of $m$ data items consistent with that element and no other. A teacher which can find such a set for each element can thus teach any element to any consistent learner with $m$ or fewer data items. This very much corresponds to the minimal definite finite tell tale sets (DFFT) from (59 - Gierasimczuk and de Jongh) for finitely identifiable classes of languages.

An alternative view concerning a *minimal adequate teacher* was introduced by (17 - Angluin). She questions the following: *How "acceptable" is the assumption of minimal adequacy of the teacher? How "feasible" are the computations required for a minimally adequate teacher in any setting?* It seems that membership items is an unobjectionable ability to require. Finding counterexamples seems a bit more problematic since this requires a very precise and explicit representation of the correct hypothesis from the teacher. Removing some of the limitations of the assumption of minimal adequacy of the teacher, should provide interesting insights of how the learner process information. This should be investigated since will account for a more realistic scenario where the teacher can make mistakes.

## 5.5 Conclusion

In this chapter we addressed five different learning environments with five different ways of presenting information. We are gradually increasing the difficulty of the learning by reducing explicit information in the positive stream of data. We showed that this difficulty factor do affects the positive results for

learnability of class $\mathcal{R}$. The fourth and fifth method were more susceptible to the increment in difficulty, since seems more difficult to identify class $\mathcal{R}$ in the limit for such cases. This suggested that information presented is not enough to *efficiently* learn an inference system from $\mathcal{R}$, even for $R_{ND}$. We proposeed some basic notions and a general supervised learning procedure for class $\mathcal{R}$. The learner starts the process of building a complete proof with his current hypothesis, procedure bounded by a given natural number. The counterexamples by means of positive and negative items will help the learner to easily exclude some alternatives.

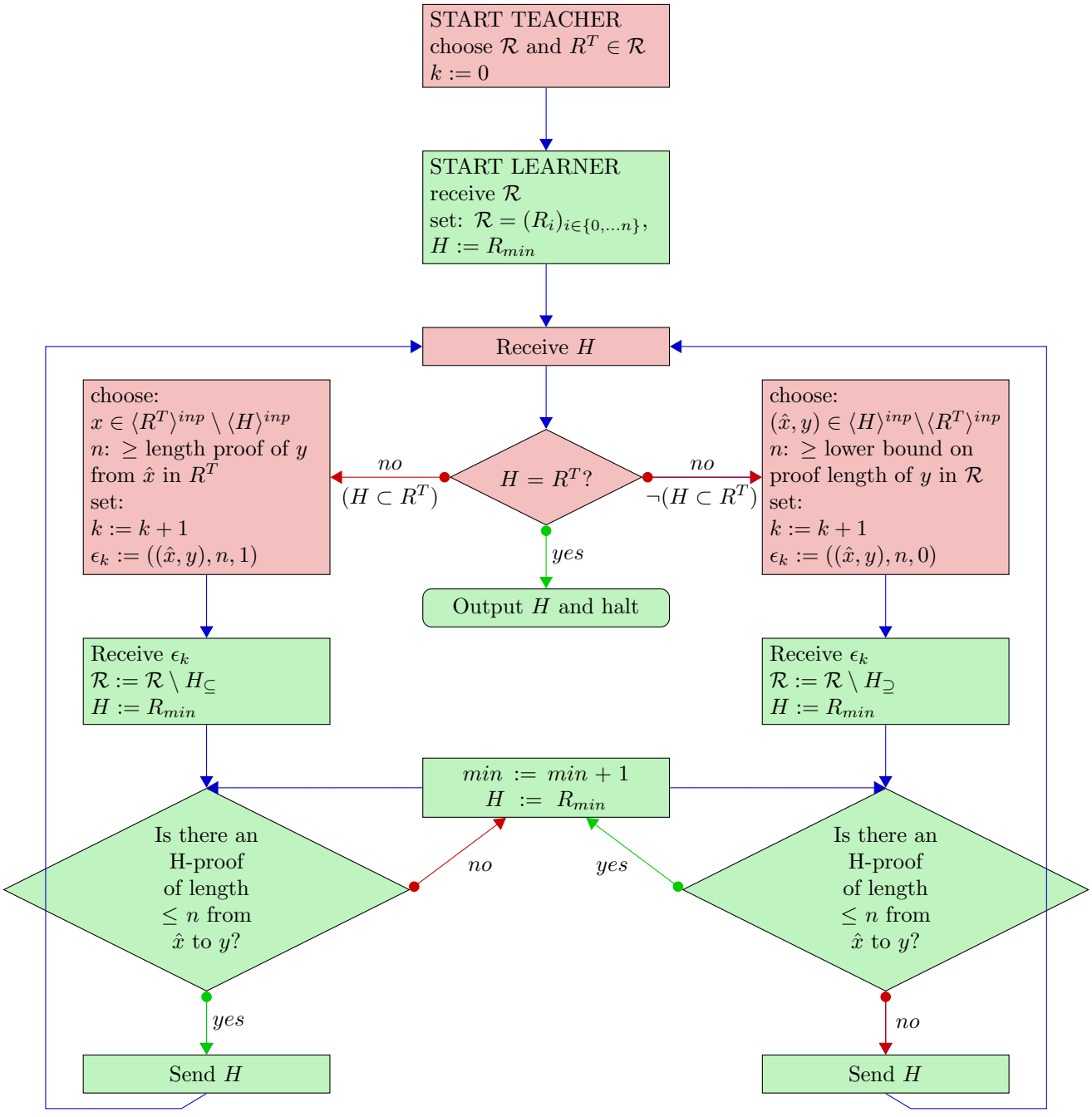


Figure 5.2: Learner and Teacher Interaction: Teacher is represented in red and Learner in green

# Chapter 6

# Results and Future work

## 6.1 Summary

The goal of our work was to define and study the properties of a formal learnability model of reasoning. Let us look back and see to what extent we have fulfilled this goal.

We started by providing the background and necessary notions from both formal learning theory and natural deduction proof system for propositional logic, in Chapters 2 and 3. In Chapter 2 we discussed our preference for Gold's learnability approach. We also argued for the importance of the formal learning theory for cognitive science. Furthermore, we also pointed out the lack of theoretical accounts for studying *learning of deductive reasoning*. In Chapter 3 we presented the usual Gentzen's style natural deduction proof system for propositional logic. Specifically, in Section 3.3 we asked what is the best way to formalize the inference systems from the perspective of our learning problem. This led to three possible ways of representing the inference rules: as grammatical rules, rules in the form of an axiom (similar to propositional formulas), and as, usually, inference rules defined within proof theory.

We concluded that a hybrid of these three forms is the best way to represent our learning space.

In Chapter 4 we presented the mathematical formalization of our framework, we provided an explicit mathematical description of the new version of natural deduction system as well as the alternative inference systems included in the learning space. We also defined proofs as finite sequences of reasoning stages. It yielded an analogy with Gold'paradigm: the inference systems were *conceived* to be similar to *grammars*; and the sets of proofs they generate can be viewed as *languages*.

In Chapter 5 we analyzed five different learning environments with five different ways of presenting arguments to the learner. We gradually increased the difficulty by reducing explicit information. We showed how such manipulation affect learnability. In particular, if the learner is only presented with *necessary premises* and *necessary open assumptions* then an *efficient* learning procedure is impossible. Thus, we turned our attention towards the more powerful framework. We introduced the basic notions for a computational model with a learner and a teacher who interact in a sequence of exchanges. In this framework, we implemented a learning model powerful enough to yield learnability result for the class escaping 'classical' learnability.

## 6.2 Future work

Even though we focused on the identification in the limit we realize that other learning paradigms could be also explored to better capture the conceptual and possibly cognitive underpinnings of the problem. For instance, one could try to approach the problem within the Valiant's learnability framework, and then, compare the results. Another interesting research direction would be to formally evaluate the complexity implications of our model. That could lead to a simpler model with bounded computational resources (c.f. 60). We believe that considering computational restrictions could bring us closer toward a cognitive computational model of learning deductive reasoning. with special emphasis on modular-cognitive-architecture frameworks such as Soar (61, 62), or ACT-R (63). These sort of models could find potential application in automated theorem proving, inductive program synthesis, intelligent pedagogical systems, etc. Finally, such models could be compared with experimental data leading to a better understanding

of cognitive processes supporting reasoning.

## 6.3 Conclusion

Our main results suggest that any alternative inference system originating from misinterpretations of the normative inference rules in logic can be acquired, or in our terms, *learned*. Furthermore, we learned that a competent teacher is necessary for learning deductive reasoning. The *right*, localized interventions from an informant (that in our model would be the interventions of the teacher) can help developing the sufficient skills in order to recognize and learn the *correct* reasoning system.

# Bibliography


[1] Nina Gierasimczuk, Han L.H Van der Maas, and Maartje EJ Raijmakers. An analytic tableaux model for deductive mastermind empirically tested with a massively used online learning system. *Journal of Logic, Language and Information*, 22(3):297–314, 2013.

[2] Andrew J.B Fugard, Niki Pfeifer, Bastian Mayerhofer, and Gernot D Kleiter. How people interpret conditionals: shifts toward the conditional event. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(3):635, 2011.

[3] Samuel R Buss. *Handbook of proof theory*. Elsevier, 1998.

[4] Sanjay Jain, Daniel Osherson, James S Royer, and Arun Sharma. *Systems that learn*, volume 2. MIT press Cambridge, 1999.

[5] Mark E Gold. Language identification in the limit. *Information and control*, 10(5):447–474, 1967.

[6] Dana Angluin and Carl H Smith. Inductive inference: Theory and methods. *ACM Computing Surveys (CSUR)*, 15(3):237–269, 1983.

[7] Kevin T Kelly. *Logic of reliable inquiry*. Oxford University Press, 1995.

[8] Lance J Rips. *The psychology of proof: Deductive reasoning in human thinking*. MIT Press, 1994.

[9] Martin D.S Braine and David P O'Brien. *Mental logic*. Psychology Press, 1998.

[10] Ray J Solomonoff. A formal theory of inductive inference. Part I. *Information and control*, 7(1): 1–22, 1964.

[11] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[12] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, 2013.

[13] Hilary Putnam. Probability and confirmation. US Information Agency, Voice of America Forum, 1964.

[14] Lenore Blum and Manuel Blum. Toward a mathematical theory of inductive inference. *Information and control*, 28(2):125–155, 1975.

[15] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[16] Dana Angluin. Inductive inference of formal languages from positive data. *Information and control*, 45(2):117–135, 1980.

[17] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987.

[18] Dana Angluin and Mārtiņš Kriķis. Teachers, learners and black boxes. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 285–297. ACM, 1997.

[19] Leonard Pitt and Manfred K Warmuth. The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of the ACM (JACM)*, 40(1):95–142, 1993.

[20] Pieter W Adriaans. Language learning from a categorial perspective. 1992.

[21] Menno Van Zaanen. ABL: Alignment-based learning. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 961–967. Association for Computational Linguistics, 2000.

[22] Alvis Brazma, Inge Jonassen, Ingvar Eidhammer, and David Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of computational biology*, 5(2):279–305, 1998.

[23] Yasubumi Sakakibara. Recent advances of grammatical inference. *Theoretical Computer Science*, 185(1):15–45, 1997.

[24] Colin De La Higuera. A bibliographical study of grammatical inference. *Pattern recognition*, 38(9): 1332–1348, 2005.

[25] Shan-Hwei Nienhuys-Cheng and Ronald De Wolf. *Foundations of inductive logic programming*, volume 1228. Springer, 1997.

[26] James Jay Horning. A study of grammatical inference. Technical report, DTIC Document, 1969.

[27] Kent Johnson. Gold's theorem and cognitive science*. *Philosophy of Science*, 71(4):571–592, 2004.

[28] Lance J Rips. Goals for a theory of deduction: Reply to johnson-laird. *Minds and Machines*, 7(3): 409–424, 1997.

[29] Philip N Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Number 6. Harvard University Press, 1983.

[30] Philip N Johnson-Laird. An end to the controversy? A Reply to Rips. *Minds and Machines*, 7(3): 425–432, 1997.

[31] Steven N Goodman. Toward evidence-based medical statistics. 2: The Bayes factor. *Annals of internal medicine*, 130(12):1005–1013, 1999.

[32] Michael C Frank and Noah D Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998, 2012.

[33] Gerhard Gentzen. Investigations into logical deduction. *American philosophical quarterly*, pages 288–306, 1964.

[34] D Prawitz. Natural Deduction: A Proof-Theoretical Study. *Almqvist and Wiksell, Stockholm*, 1965.

[35] Stanisław Jaśkowski. *On the rules of suppositions in formal logic*. Nakładem Seminarium Filozoficznego Wydziału Matematyczno-Przyrodniczego Uniwersytetu Warszawskiego, 1934.

[36] James W Garson. Expressive power and incompleteness of propositional logics. *Journal of philosophical logic*, 39(2):159–171, 2010.

[37] Martin D.S Braine, Brian J Reiser, and Barbara Rumain. Evidence for the theory: Predicting the difficulty of propositional logic inference problems. *Mental logic*, pages 91–144, 1998.

[38] Daniel N Osherson. *Logical abilities in children: IV. Reasoning and concepts.* Lawrence Erlbaum Assoc, 1976.

[39] William Bechtel and Robert C Richardson. *Discovering complexity: Decomposition and localization as strategies in scientific research.* MIT Press, 2010.

[40] Jerry A Fodor. *The language of thought*, volume 5. Harvard University Press, 1975.

[41] Jerold A Fodor and Merrill F Garrett. The psychological unreality of semantic representations. *Linguistic Inquiry*, pages 515–531, 1975.

[42] Francis J Pelletier and Allen P Hazen. A history of natural deduction. *Logic: A History of Its Central Concepts*, 11:341–414, 2012.

[43] Henk Barendregt and Silvia Ghilezan. Lambda terms for natural deduction, sequent calculus and cut elimination. *Journal of Functional Programming*, 10(01):121–134, 2000.

[44] Niki Pfeifer and Gernot D Kleiter. Coherence and nonmonotonicity in human reasoning. *Synthese*, 146(1-2):93–109, 2005.

[45] Richard Zach. Completeness before post: Bernays, Hilbert, and the development of propositional logic. *Bulletin of Symbolic Logic*, 5(03):331–366, 1999.

[46] Anne S Toelstra. Scwichtenberg, basic proof theory. *Cambridge Tracts in Theoretical Computer Science, CUP*, 2000.

[47] Dirk Van Dalen. *Logic and structure*, volume 3. Springer, 1994.

[48] Douglas L Medin, Norbert Ross, Scott Atran, Russell C Burnett, and Sergey V Blok. Categorization and reasoning in relation to culture and expertise. *Psychology of learning and motivation*, 41:1–42, 2002.

[49] John E Hopcroft and Jeffrey D Ullman. Formal languages and their relation to automata. 1969.

[50] Peter C Wason. Reasoning about a rule. *The Quarterly journal of experimental psychology*, 20(3): 273–281, 1968.

[51] Leda Cosmides. The logic of social exchange: Has natural selection shaped how humans reason? studies with the wason selection task. *Cognition*, 31(3):187–276, 1989.

[52] Amos Tversky and Daniel Kahneman. Extensional versus intuitive reasoning: the conjunction fallacy in probability judgment. *Psychological review*, 90(4):293, 1983.

[53] Niki Pfeifer and Gernot D Kleiter. The conditional in mental probability logic. *Cognition and conditionals: Probability and logic in human thought*, pages 153–173, 2010.

[54] Anne S Toelstra and D van Dalen. Constructivism in mathematics. an introduction. *Studies in Logic and the foundation of Mathematics*, 121.

[55] Orly Buchbinder and Orit Zaslavsky. How to decide? students' ways of determining the validity of mathematical statements. In *Proceedings of the Fifth Congress of the European Society for Research in Mathematics Education*, pages 561–570, 2007.

[56] Orly Buchbinder and Orit Zaslavsky. Is this a coincidence? the role of examples in fostering a need for proof. *ZDM*, 43(2):269–281, 2011.

[57] Dana Angluin and Leonor Becerra-Bonache. Effects of meaning-preserving corrections on language learning. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 97–105. Association for Computational Linguistics, 2011.

[58] Sally A Goldman and David H Mathias. Teaching a smart learner. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 67–76. ACM, 1993.

[59] Nina Gierasimczuk and Dick de Jongh. On the minimality of definite tell-tale sets in finite identification of languages. *Logic and Interactive Rationality*, page 21, 2009.

[60] Claes Strannegård, Abdul R Nizamani, Anders Sjöberg, and Fredrik Engström. Bounded kolmogorov complexity based on cognitive models. In *Artificial General Intelligence*, pages 130–139. Springer, 2013.

[61] Paul S Rosenbloom, John Laird, and Allen Newell. The soar papers: Research on integrated intelligence. 1993.

[62] Virginia A Peck and Bonnie E John. Browser-soar: A computational model of a highly interactive task. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 165–172. ACM, 1992.

[63] John R Anderson, Michael Matessa, and Christian Lebiere. Act-r: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, 12(4):439–462, 1997.