CHAPTER 1

INTRODUCTION AND OVERVIEW

— in which the motivation for this study will be given, its scope will be delineated, preliminary concepts and notation will be defined, and an overview will be provided of what is still to come —

§1. MOTIVATION AND SCOPE OF THE THESIS

THIS THESIS GIVES an account of my investigations into the logical foundations of inductive reasoning. Roughly speaking, induction consists in identifying the similarities between observed objects or events, and hypothetically extending those similarities to unobserved objects or future events. It constitutes a mode of reasoning which plays a role in the acquisition of scientific theories, as well as the acquisition of the descriptive vocabulary needed to reason about the objects and events encountered in everyday life, although, it must be added, philosophers (and psychologists) disagree about the importance of induction in these processes. In recent years researchers in artificial intelligence have explored the realisation of inductive procedures by means of computer programs, not only with the purpose to automate the formation of scientific theories and concept taxonomies, but also aiming at extending inductive techniques to new tasks, such as the construction of computer programs from examples of their intended behaviour (like in inductive logic programming). The initial motivation for the present study grew out of a perceived need for a formal framework in which the various inductive techniques that are used in practice can be precisely characterised and related to each other. As such, it should be viewed as a methodological investigation into the formal aspects of inductive methods as used in computer science in general, and artificial intelligence in particular.

The field of artificial intelligence seeks to implement aspects of human cognitive behaviour by means of computer programs. By its very nature, the field has a strong inclination towards epistemological subjects, as they were traditionally studied by philosophers and logicians[1]. Clearly, researchers in each of these fields study the same subjects with quite different aims. For instance, philosophers are interested in such issues as the scope and justification of human knowledge; logicians are interested in formalising reasoning processes; and computer scientists may use techniques similar to human reasoning methods to solve particular problems. Despite these different aims, methods and results from one of these fields may be relevant for the others. For instance, Gödel's proof of the incompleteness of first-order predicate logic as soon as it includes fundamental

[1]The psychological aspects of induction fall outside the scope of this thesis.

number theory has some serious philosophical implications; and the computer scientist's pursuit for an efficiently implementable deduction strategy has spawned an innumerable amount of logical investigations. Many problems in artificial intelligence would therefore benefit from an approach that is not completely ignorant of related views, and inductive reasoning is no exception in this respect. In this thesis inductive reasoning is studied in a multidisciplinary context, combining perspectives from philosophy, logic, and artificial intelligence.

Philosophy

Philosophical investigations of inductive reasoning have concentrated on the justification of inductively acquired knowledge: the infamous 'Problem of Induction'. Not only is this a very complex problem, philosophers disagree about what exactly should be understood as a justification. Many scholars tend to believe that this requires, in some form or another, an assessment of the truth of the inductive conclusion, given the truth of the premisses. Other philosophers claim that describing the inferential patterns underlying induction suffices. For instance, Nelson Goodman states that 'the basic task in justifying an inductive inference is to show that it conforms to the general rules of induction' (Goodman, 1954, p.374). Such rules of induction delimit what kind of argument we are prepared to accept as inductive argument — they cannot have, and do not need, any further justification than their compliance with common practice. Furthermore, as Goodman notes, this renders the inductive justification problem completely analogous to the justification of deduction: there, also, a deductive argument is valid if it is in accordance with the rules of the game — rules that come very close to formalising our intuitions about deduction, it is true, but which nevertheless can also only be justified by an appeal to intuition.

It would be, I believe, a mistake to think that the 'Problem of Induction' can be completely reduced to the problem of devising a proper set, or proper sets, of rules of induction, because a solution to the latter problem leaves unanswered the question how to assess the truth of the inductive conclusion. The two problems — justifying inductive reasoning and describing inductive reasoning — are, in the view developed in this thesis, relatively unrelated. Furthermore, I think that the justification problem is not reserved for induction, but manifests itself in any form of non-deductive reasoning. In this thesis I will concentrate on the description problem rather than the justification problem, although I will also spend some words on the latter.

Logic

As a separate branch of science, logic has emerged only relatively recently. Until the twentieth century questions regarding the patterns underlying human reasoning were considered to belong to philosophy. This century has seen a tremendous breakthrough in the formalisation of mathematical reasoning, and as a result logical investigations have become more technical and less philosophical. Logic, as we know it today, is a technical discipline with the mathematics of deductive logic as its main subject.

Without questioning the worth of current-day logical investigations, I consider it regrettable that the original, general question regarding the patterns underlying human

reasoning has been overpowered by a rather more specific question concerning the logical patterns of deductive reasoning. It is true that non-deductive forms of reasoning can never be formalised to the same extent as deductive reasoning, for the simple reason that the latter has a built-in notion of 'correct' reasoning which the former lack. However, I believe that some present-day logical tools can be successfully used to obtain a deeper understanding of non-deductive forms of reasoning, and this thesis is meant as a contribution in this direction.

Artificial intelligence

If logic is a young field, artificial intelligence is still in its infancy. Artificial intelligence can be described as the field that aims to automate human cognitive abilities, in order to improve the usefulness of computers[2]. Artificial intelligence research sheds a new light on many old philosophical problems by taking a computational viewpoint. For instance, many philosophers, including Peirce and Popper, think that scientific hypotheses come to us in a 'flash of insight', and not by an algorithmic process consisting of small reasoning steps. There is however, as Peirce noted, a certain relation between the hypothesis and the observations preceding it, parts of which can be logically formalised, and such a formal relation can be computed[3]. While the tendency in philosophy has been 'humans do not do this algorithmically, so don't bother about the logical relation', the artificial intelligence attitude is 'but I want my computer to do it algorithmically, so I need the exact logical relation as a specification for the computer program'.

We could say that artificial intelligence researchers are philosophical programmers (or that philosophers are artificial intelligence designers). The interplay between the two disciplines is likely to produce new insights, and this thesis is hoped to contribute in that respect.

§2. PRELIMINARIES

In this section I will introduce the most important concepts, notation and terminology used throughout the thesis. A summary of many of these terms can be found in an appendix.

Terminology

If it were possible, at this stage, to give a precise, coherent and undisputed definition of the terms used in the following chapters, this thesis wouldn't have been written. For instance, there is no well-established definition of induction, and the interpretation of the term 'abduction' is a battlefield. The following discussion of the terminology I use in this thesis is therefore partly premature and subjective.

---

[2] I regard artificial intelligence as a subfield of computer science rather than cognitive science — for a collection of views on this subject see (Flach & Meersman, 1991).

[3] It may have to be approximated, or limited to special cases, because of undecidability or complexity problems, but this leaves the main point unaffected.

1. Introduction and overview

Reasoning is an informal term denoting the process of forming arguments, i.e. drawing conclusions from premisses, and *logic* is the formal study of that process. It is taken for granted that different *reasoning forms* can be identified, such as inductive reasoning, deductive reasoning, plausible reasoning, and so forth; again, this term will be used informally. The formalisation of different reasoning forms and their relations is seen as the main goal of logic, resulting in a catalogue of reasoning forms; such a catalogue, or a coherent part of it, will be referred to as a *descriptive logical theory*[4].

My definition of the deductive reasoning form is rather generous. An argument is *deductive* if the conclusion cannot be contradicted by new knowledge without contradicting the premisses also; a form of reasoning is deductive if it only allows deductive arguments. Another way to say the same thing is: deduction is the logic of *non-defeasible reasoning*. This is not meant to say that there is a single deductive logic, and that it is clear which arguments are deductively valid and which are not. On the contrary: the argument 'two plus two equals four; therefore, if the moon is made of green cheese, then two plus two equals four' will be rejected by those who favour a causal or relevance interpretation of if–then rather than a truth-functional interpretation. However, as soon as such an argument is accepted as deductively valid, the only way to defeat the conclusion is by denying that two plus two equals four, and this defeats the premisses also. Note that I didn't talk about the logical language, or about the proposed semantics: modal, temporal, relevance, and intuitionistic logics are all formalisations, sometimes conflicting, of certain aspects of deductive reasoning.

Non-deductive reasoning forms are defeasible: a conclusion may be defeated by new knowledge, even if the premisses on which the conclusion was based are not defeated. For instance, the argument 'birds typically fly; Tweety is a bird; therefore, Tweety flies' is non-deductive, since Tweety might be an ostrich, hence non-typical. The argument 'every day during my life the sun rose; I don't know of any trustworthy report of the sun not rising one day in the past; therefore, the sun will rise every future day' is non-deductive, since if the sun would not rise tomorrow, this would invalidate the conclusion but not the premisses.

The Tweety-argument is a well-known example of what I call *plausible reasoning*: reasoning with general cases and exceptions. This terminology is not generally accepted: this form of reasoning is normally referred to as non-monotonic reasoning[5]. A reasoning form is monotonic if, given an argument, adding a premiss cannot defeat the conclusion. In fact, this is the same property as what I called non-defeasibility above; since any non-deductive reasoning form is defeasible, it follows that any non-deductive reasoning form is non-monotonic. In other words, the property of non-monotonicity is of limited use in

[4] I don't think that the field of logic, in its current state, has developed very well towards this goal. Most of the logics around, such as modal, temporal, partial and relevance logics, are mostly variations upon a theme, the theme of deductive reasoning, and thus represent only a tiny subspectrum of the huge range of possible reasoning forms. Like algebra has generalised the properties of numbers into such abstract concepts as groups, rings and fields, logic should investigate what properties of deductive logic are contingent and could be different, and what properties are tied to the nature of logic, expressing an inherent quality of reasoning. Such investigations belong to the discipline of descriptive logic.

[5] *Default reasoning* would be a good term, but this seems too strongly connected to a particular logic, i.e. default logic.

classifying reasoning forms; for this reason I prefer a different (and more meaningful) term. Typically, plausible reasoning encompasses deductive reasoning, but also tries to draw, in the absence of crucial information, conclusions that are not deductively justified. In this sense plausible reasoning is 'supra-deductive' or, as I will call it, *quasi-deductive*.

The second argument above, concerning the prediction of sunrise, is an example of *induction*, which is commonly defined as reasoning from specific *observations* (also called evidence) to general rules or *hypotheses*. As a first attempt this is an acceptable definition, but note that it leaves the logical relation between observations and inductive hypothesis unspecified. After all, after observing 100 white swans we might conclude that swans may have any colour. Few people would accept this inductive conclusion, but why is this so? Like with deductive reasoning this should be based on some notion of 'inductive validity'. This is exactly the subject of this thesis, although I will eschew the term 'validity' because of its strong deductive connotation. Note that inductive reasoning does not comprehend all deductively valid arguments and is therefore not quasi-deductive; I will call reasoning forms that do not aim at approximating deduction *a-deductive*.

Abduction is a term originally introduced by C.S. Peirce to denote the process of forming an explanatory hypothesis given some observations (a hypothesis from which the observations can be deduced). According to the view defended in this thesis, inferring a general explanation of observations is one possible form of inductive reasoning, so we might say that abduction is a special case of induction. However, in recent years a different notion of abduction has emerged in the logic programming field, according to which the general explanation is known, but one of its premisses is not known to be true; abduction is then seen as hypothesising this missing premiss. As a consequence, abduction and induction are viewed as complementary: induction infers the general rule, given that its premisses and its conclusion hold in specific cases; abduction infers specific premisses, given the general rule, and specific instances of its conclusion and some of its premisses. In this thesis I will stick to the former interpretation of abduction, as originally intended by Peirce; in order to minimise confusion I will mostly avoid the term abduction in favour of the term *explanatory reasoning*.

Explanatory reasoning is meant to formalise one aspect of inductive reasoning, namely that inductive hypotheses should be able to explain the observations. *Confirmatory reasoning* formalises another intuitive aspect of induction, that is, the idea that the inductive conclusion should be confirmed by the hypothesis. One might expect that the ideal inductive hypothesis is both explanatory and confirmed; however, straightforward logical formalisations of both aspects turn out to interfere in such a way that they have been developed separately in this study. Neither of these formalisations is intended to fully capture the essence of inductive reasoning; therefore, I tend to avoid the term induction in the more formal parts of the thesis, and adopt the more neutral term *conjectural reasoning*; the term *conjecture* is used synonymously with 'hypothesis' in the sense of 'defeasible general rule'.

Logical preliminaries

Throughout the thesis I assume that a logical language L is fixed. Except for chapter 7, whose completeness results only hold if L is a (first-order) propositional language, I will

normally assume that L is a (first-order) predicate-logical language, unless indicated otherwise. Formally, such a language is defined by a denumerable set of predicate symbols with arity 0,1,2,…, a denumerable set of variable symbols, a denumerable set of constant symbols, and a denumerable set of function symbols with arity 1,2,…. Terms and well-formed formulas (wff's) are defined in the usual recursive way, by means of the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$, and the quantifiers $\forall$ and $\exists$:

1a. a constant symbol is a term;
1b. a variable symbol is a term;
1c. if f is a function symbol with arity n, and $t_1,\ldots,t_n$ is a sequence of n terms, then $f(t_1,\ldots,t_n)$ is a term;
1d. nothing else is a term.
2a. **`true`** and **`false`** are wff's;
2b. if P is a predicate symbol with arity n, and $t_1,\ldots,t_n$ is a sequence of n terms, then $P(t_1,\ldots,t_n)$ is a wff, also referred to as an *atomic formula*, or briefly, an *atom*;
2c. if F_1 and F_2 are wff's, then $(\neg F_1)$, $(F_1 \wedge F_2)$, $(F_1 \vee F_2)$, $(F_1 \rightarrow F_2)$, and $(F_1 \leftrightarrow F_2)$ are wff's;
2d. if x is a variable and F is a wff, then $(\forall x\colon F)$ and $(\exists x\colon F)$ are wff's;
2e. nothing else is a wff.

Brackets will usually be dropped as much as possible without causing confusion. A *propositional language* is a special case of a predicate-logical language, built only from predicate symbols with arity 0, referred to as *proposition symbols* or *propositional atoms*, and connectives; the set of propositional wff's is defined by clauses 2a–c and 2e.

The *scope* of a quantifier $\forall x$ (resp. $\exists x$) in $\forall x\colon F$ (resp. $\exists x\colon F$) is F. An occurrence of a variable in a formula is *bound* if it immediately follows a quantifier, or if it occurs in the scope of a quantifier with the same variable. Any other occurrence of a variable in a formula is *free*. Although the same variable can have both bound and free occurrences within a formula, this is usually avoided by renaming; in that case we simply refer to free and bound variables. A *closed* formula is a formula without free occurrences of any variable; otherwise the formula is *open*. For any formula, $\forall(F)$ denotes the *universal closure* of F, which is the closed formula obtained by adding a universal quantifier for every variable with a free occurrence in F.

If A is an atom, then $\neg A$ is a *negative literal*; an atom is also referred to as a *positive literal*. A *clause* is the universal closure of a disjunction of positive and negative literals. We adopt the usual notation for clauses: a clause like $A_1\texttt{;}A_2\texttt{:-}B_1\texttt{,}B_2$ stands for the formula $\forall(A_1 \vee A_2 \vee \neg B_1 \vee \neg B_2)$; the part preceding the symbol '`:-`' is called the *head* of the clause, the part succeeding it is called the *body*. A clause is a *denial* if it has no positive literal, *definite* if it has one positive literal, and *indefinite* if it has more than one positive literal. A non-definite clause is also called an *integrity constraint*. A *fact* is a definite clause without negative literals; the '`:-`' symbol is omitted for facts. A *normal* clause is obtained from an indefinite clause by moving all but one positive literals to the body, preceded by a negation symbol. A *logic program* is a set (conjunction) of definite and normal clauses.

We conform to the Prolog notation (see Flach, 1994): predicate symbols, function symbols and constant symbols start with a lowercase letter, while variable symbols start

with an uppercase letter. A *functor* is a function symbol occurring in a clause. The term `.(X,Y)`, where '`.`' is the *list functor*, is usually written `[X|Y]`; the *empty list constant* is `[]`, and for lists of fixed length such as `.(a,.(b,.(c,[])))` we employ the linear notation and write `[a,b,c]`. A *substitution* is a function mapping variables to terms; *applying* a substitution to a clause replaces all occurrences of the variables in the domain of the substitution with the corresponding term. A *ground* term (clause) is a term (clause) without variables. The *Herbrand universe* of a language or a program is the set of ground terms. The *Herbrand base* of a language or a program is the set of ground atoms.

A *predicate-logical interpretation* is a pair $\langle D,i\rangle$, where D is a non-empty *domain* of individuals, and i is a function assigning to every constant symbol an element of D, to every function symbol with arity n a mapping from D^n to D, and to every predicate symbol with arity n a mapping from D^n to the set of *truthvalues* {**true**,**false**}[6]. A *valuation* is a function v assigning to every variable an element of D. Given an interpretation $I = \langle D,i\rangle$ and a valuation v a mapping i_v from terms to individuals and from formulas to truthvalues is defined as follows:

1a. if t is a constant symbol, $i_v(t) = i(t)$;
1b. if t is a variable symbol, $i_v(t) = v(t)$;
1c. if t is a term $f(t_1,\ldots,t_n)$, $i_v(t) = i(f)(i_v(t_1),\ldots,i_v(t_n))$.
2a. $i_v($**true**$)$ = **true**; $i_v($**false**$)$ = **false**;
2b. if F is an atom $P(t_1,\ldots,t_n)$, $i_v(F) = i(P)(i_v(t_1),\ldots,i_v(t_n))$;
2c. $i_v(\neg F)$ = **true** if $i_v(F)$ = **false**, and **false** otherwise;
$i_v(F_1\wedge F_2)$ = **true** if $i_v(F_1)$ = **true** and $i_v(F_2)$ = **true**, and **false** otherwise;
$i_v(F_1\vee F_2) = i_v(\neg(\neg F_1\wedge\neg F_2))$;
$i_v(F_1\rightarrow F_2) = i_v(\neg F_1\vee F_2)$;
$i_v(F_1\leftrightarrow F_2) = i_v((F_1\rightarrow F_2)\wedge(F_2\rightarrow F_1))$;
2d. $i_v(\forall x\colon F)$ = **true** if $i_{v_{x\rightarrow d}}(F)$ = **true** for all $d\in D$, and **false** otherwise; where $v_{x\rightarrow d}$ is

An interpretation I *satisfies* a formula F, notation $I \models F$, if $i_v(F)$ = **true** for all valuations v; we usually say that I is a *model* of F. Models are usually denoted by the letter m, and occasionally treated as functions assigning truthvalues to formulas. A formula is *satisfiable* if it is satisfied by some interpretation, *unsatisfiable* otherwise. If all models of a set of formulas Σ are also models of φ, we say that Σ *logically entails* φ or φ is a *logical consequence* of Σ, and write $\Sigma \models \varphi$[7]. If Σ is empty, φ is called a *tautology*; instead of $\varnothing \models \varphi$, we write $\models \varphi$. A formula ψ is a *contradiction* if $\neg\psi$ is a tautology. In chapter 7 the following *completness* property of first-order predicate logic will be used: a set of formulas Σ is satisfiable iff every finite subset of Σ is satisfiable.

[6] I will not distinguish between the symbol in the language denoting a truthvalue and the truthvalue itself. Note that in §28 we consider *partial* or three-valued interpretations; the main differences with two-valued interpretations will be explained there.

[7] Conforming to standard logical practice the symbol $\models$ denotes both the satisfaction relation between interpretations and formulas, and the entailment relation between sets of formulas and formulas. Wherever this might cause confusion the intended meaning is indicated in words.

1. Introduction and overview

A *Herbrand interpretation* is an interpretation $\langle D,i\rangle$ in which the domain D is the Herbrand universe H of ground terms, while i maps every constant symbol to itself, and every function symbol f with arity n to a function from H^n mapping a sequence of ground terms $\langle t_1,\ldots,t_n\rangle$ to $f(t_1,\ldots,t_n)$. Thus, Herbrand interpretations only differ in their truthvalue assignments to predicate symbols; consequently, any Herbrand interpretation is completely defined by, and usually identified with, the set of ground atoms it assigns the truthvalue **`true`** (i.e. a subset of the Herbrand base)[8]. The main differences between Herbrand interpretations and the general case are:

- the treatment of equality: syntactically different ground terms denote different individuals, and
- quantification, which is only over the known individuals: e.g. if the Herbrand universe is $\{a\}$, then $\{P(a), \exists x: \neg P(x)\}$ does not have a Herbrand model (although it is satisfiable).

For clauses, which don't contain existential quantifiers, Herbrand interpretations are sufficient, in the following sense: a set of clauses is unsatisfiable iff it does not have a Herbrand model.

A *proof procedure* consists of a set of (logical) axioms and inference rules. Given a proof procedure P, we say that φ is *provable* from Σ and write $\Sigma \vdash_P \varphi$ if there exists a finite sequence of formulas $\varphi_1, \varphi_2, \ldots, \varphi_n$ which is obtained by successive application of inference rules to axioms, previous formulas in the sequence, or combinations of these, while φ_n is the conclusion φ. Such a sequence of formulas, if it exists, is called a *proof* of φ from Σ. A proof procedure is *sound* with respect to the semantics established by predicate-logical interpretations if $\Sigma \vdash_P \varphi$ whenever $\Sigma \models \varphi$; it is *complete* if $\Sigma \models \varphi$ whenever $\Sigma \vdash_P \varphi$. For a sound and complete proof procedure for first-order predicate logic, see e.g. (Tärnlund 1984, p.1). A set of formulas Σ is *consistent* with respect to a proof procedure P, if not both $\Sigma \vdash_P \varphi$ and $\Sigma \vdash_P \neg\varphi$ for some formula φ; two sets of formulas are *compatible* if their union is consistent. I will usually omit the reference to a particular proof procedure, and write $\vdash$ for some provability relation that coincides with $\models$. Likewise, if a set of formulas is called consistent, a sound and complete proof procedure is understood; consequently, a set of formulas is consistent iff it is satisfiable.

In chapter 8 I will make use of the proof procedure of resolution in clausal logic. In clausal logic no interaction occurs between the connectives due to the normal form in which clauses are written, and thus the set of axioms is empty. The single inference rule of *resolution* allows one to infer, from two clauses $F_1 \vee L_1$ and $F_2 \vee \neg L_2$, the clause $(F_1 \vee F_2)\theta$, where θ is the *most general unifier* of L_1 and L_2 (the minimal substitution such that $L_1\theta = L_2\theta$). In terms of definite clauses this amounts to matching the head of one clause with a literal in the body of another. The resolution proof procedure is sound and complete, but it is *refutation-complete*, i.e. if a set of clauses is inconsistent then resolution is able to derive the unsatisfiable empty clause $\square$. The [illegible] conjunction of positive literals, are transformed to refutation proofs $\Sigma \cup \{\neg\varphi\} \vdash \square$ where $\neg\varphi$ is a denial called a *query* and written $?\text{-}B_1, \ldots, B_n$. As we saw earlier, Herbrand

[8] The same observation can be made for propositional interpretations.

interpretations are sufficient to characterise such a refutation proof procedure in clausal logic. *SLD resolution* is a particular resolution proof procedure specifically tailored for definite logic programs; *SLDNF resolution* is an extension of SLD resolution that is able to deal with normal clauses by *negation as failure*.

§3. OVERVIEW OF THE THESIS

The thesis consists of three parts: Backgrounds, Foundations, and Practice. A brief overview of the chapters in each of these parts follows.

Backgrounds

In the first part work in philosophy, logic, and artificial intelligence that is relevant for the present investigations is reviewed. The choice of works is subjective, and no claim is made as to the completeness of these overviews. On the contrary, I often concentrate on one or two authors whose work has inspired mine. Apart from some minor terminological adjustments[9], which I have permitted myself with a view to overall consistency, I have tried to remain as faithful as possible to the original author's views and intentions. Where I have felt the need to comment on or voice disagreement with those views I have done so in a separate discussion section at the end of the chapter.

In the first chapter in this part, *The philosophy of induction*, I discuss the philosophical backgrounds of this study. I concentrate on philosophers who have studied induction from a logical perspective, most notably Peirce and Hempel. The work of Carnap on confirmation measures is briefly reviewed in the discussion section, since it provokes some reflections on the aims and scope of logic (further dealt with in chapter 5). However, his numerical approach is not seen as fundamental to the subject of this thesis, since it does not provide any insight into the **logic** of induction. The main conclusion drawn from this chapter is that the dichotomy between explanatory and confirmatory induction proposed and defended in this thesis is already present, albeit implicit, in the work of Peirce and Hempel.

The next chapter is called *Approaches to computational induction*. It draws upon work in machine learning (a subfield of artificial intelligence) on inductively learning concepts, logic programs, and logical theories from examples. I indicate how the latter two problems can be reformulated as problems of explanatory and confirmatory induction, respectively.

The third and final chapter in the first part, *The analysis of non-deductive reasoning*, is mainly devoted to one article by Kraus, Lehmann and Magidor that provided the main inspiration for my approach. In that article the authors set out to "study general patterns of nonmonotonic reasoning". The main tool they used for that study is the notion of a consequence relation, which is a set of pairs of premiss and conclusion, originally proposed by Gabbay. By formulating and combining properties of such consequence relations, such as (Cautious) Monotonicity and Transitivity, they succeeded in providing a systematic and precise overview of different forms of plausible reasoning, that is, a

[9] all of which are indicated in footnotes.

descriptive theory of plausible reasoning. In this thesis I have set out to do the same for inductive reasoning.

Foundations

The second part makes up the core of this thesis. It consists of three, increasingly technical, chapters. The first chapter, *Outline of a descriptive theory of induction*, is probably the most speculative among them, challenging some established dogmas of logic. Specifically, I take issue with the idea that a logical semantics is necessarily based on the notion of truth. In my view, any quality worth preserving in an argument may give rise to a logic formalising some useful form of reasoning. Two of the qualities that may be preserved in inductive arguments are explanatory power and regularity of interpretations[10]. Since the preservation of such qualities may not be reserved for inductive reasoning, nor characterise it completely, I introduce the term 'conjectural reasoning' for any form of reasoning involving uncertain hypotheses, including induction.

In the next chapter, *Properties of conjectural consequence relations*, I commence my study of general patterns of conjectural reasoning in the spirit of Kraus *et al*. Starting with the adequacy conditions for a material definition of the relation of confirmation formulated by Hempel, I propose a number of rules for explanatory and confirmatory reasoning, and some rules that are meaningful in both cases. Systems of such rules are studied and semantically characterised in the final chapter in this part, *Rule systems for conjectural reasoning*.

Practice

Of the two forms of induction studied in this thesis, explanatory induction and confirmatory reasoning, the latter is certainly much less understood than the former. In the third and last part of this thesis I illustrate the practice of confirmatory induction in the context of relational databases. This chapter also illustrates my claim that in confirmatory induction one needs an additional goal which the inductive hypothesis is meant to fulfil, since 'being confirmed' is not an end in itself. Those readers who wish to have a concrete idea of confirmatory induction could read this chapter before diving into the formalities of the Foundations part.

The thesis is ended with a chapter recapitulating the main achievements and conclusions, and with a few appendices including a glossary of terms and logical rules.

*

[10] A third one, generality, is briefly considered but not worked out in this thesis.