Chapter 1

Introduction

Adaptive logics are loosely characterized (Section 1.1) and some examples are presented (Section 1.2) to illustrate the dynamic reasoning forms explicated by adaptive logics. Three distinctive features are presented in Section 1.3: that adaptive logics represent a qualitative approach, that they are formal logics, and that they have (dynamic) proofs that explicate actual reasoning. In Section 1.4, the notational conventions that will be followed in this book are outlined. Sections 1.5 and 1.6 introduce the proof theory and semantics of logics that have static proofs. These are not adaptive logics but logics of the kind that was most commonly studied hitherto. As **CL** (Classical Logic) plays an important role in this book, a specific presentation of it is presented in Section 1.7.

1.1 What is an Adaptive Logic?

A *logic* is a mapping that assigns to every *premise* set a set of *consequences*. In technical terms, a logic **L** is a function that maps every set of closed formulas to a set of closed formulas—see Section 1.4 on closed formulas. So, where $\mathcal{W}$ is the set of closed formulas of the considered language and $\wp(\mathcal{W})$ is the power set of $\mathcal{W}$ (the set of all subsets of $\mathcal{W}$), $\mathbf{L}\colon \wp(\mathcal{W}) \to \wp(\mathcal{W})$. As we shall see later, a logic may be presented in very different ways.

A logic as defined here concerns a consequence relation, not a set of logical truths (theorems or valid formulas—the terms are introduced later in this chapter). A logic determines a set of logical truths, but a set of logical truths does not determine a logic. Logical truths are a side effect of the consequence relation.

When is a logic adaptive? I shall start with a loose and intuitive specification in this chapter and, in Chapter 4, shall offer a strict definition, which is called the standard format. That definition is intended as a hypothesis, viz. that all defeasible reasoning forms can be characterized by an adaptive logic in standard format. For now, however, let us move on to the loose specification.

A logic is adaptive if it adapts itself to the specific premises to which it is applied. The previous sentence will make many people frown. Being used to so-called Tarski logics (see Section 1.5), most of them may find it hard to imagine that a logic might have the property to adapt itself to a premise set. And yet, an impressive number of reasoning processes adapt themselves to their premises. Moreover, we are all familiar with them: they occur frequently both in the sciences and in everyday situations.

That the consequence set of Γ *depends* on Γ holds for nearly all logics. There are a few exceptions, but these are not very interesting anyway.[1] In order to be adaptive a logic needs a further distinctive trait: *the logic* adapts itself to the premises.

Given that a logic is a function, as stated above, that a logic adapts itself to the premises can only pertain to the presentation of the logic, for example to its proof theory or to its semantics. Thus the presence of certain premises may prevent the derivability of a formula from other premises. Also the presence of a certain subformula in a premise A may prevent the derivability of another subformula of A. Thus there are adaptive logics according to which p is derivable from $(p \wedge \neg q) \wedge r$, but not from $(p \wedge \neg q) \wedge q$—read $A \wedge B$ as A-and-B and read $\neg A$ as not-A.

Let us consider some simple examples of logics that are adaptive. Later in this chapter, I shall consider some reasoning processes that may be explicated by an adaptive logic. For now, let us concentrate on simple formal properties of logics, without worrying too much about possible applications. Let **CL** henceforth denote Classical Logic. Consider the logic, call it **L1**, that assigns A as a consequence to Γ iff (if and only if) A is a **CL**-consequence of a member of Γ that does not contradict itself. Thus p is a **L1**-consequence of $\{p \wedge q, r\}$, but p is not a **L1**-consequence of $\{p \wedge (q \wedge \neg q), r\}$ because p is not a **CL**-consequence of r whereas $p \wedge (q \wedge \neg q)$ contradicts itself. One may apply a logic like **L1** for handling (somewhat roughly) witnesses' testimonies. Every testimony is taken as a single premise—the conjunction of the statements that make up the testimony. A statement is a **L1**-consequence of the premise set iff it is a **CL**-consequence of the testimony of a witness who did not contradict herself. No **L1**-consequence of any premise set is contradictory. If the witnesses contradict each other, the **L1**-consequence set is inconsistent but not trivial. So it may be taken as a sensible basis for finding out the truth about the matter that the testimonies are about. The **CL**-consequence set does not form a sensible basis in this case because it is trivial.[2]

A very different logic, call it **L2**, displays the following behaviour. If the premise set Γ is consistent,[3] the **L2**-consequences of Γ are identical to the **CL**-consequences of Γ; if the premise set is inconsistent, and hence the set of its **CL**-consequences is trivial, the **L2**-consequences of Γ are identical to the **CLuN**-consequences of Γ—**CLuN** is a paraconsistent logic,[4] which will be introduced in Chapter 2. The adaptive logic **L2** assigns a non-trivial consequence set to all premise sets, except for a few exceptions.[5] However, **L2** belongs to a border-

[1] The exceptions are *constant* functions $\mathbf{L}\colon \wp(\mathcal{W}) \to \wp(\mathcal{W})$, which do not seem very useful as logics. This holds even for the more sensible ones, among them those that assign to Γ the empty set, or the set of all formulas, or the set of all theorems of the logic. Still, some such systems serve a purpose, as may be seen from Section 8.4.

[2] In the present context, that a set of formulas is trivial means that the set contains all formulas. In Section 1.7 I introduce a more usual definition which relates the triviality of a set of formulas to a logic. On both readings, every inconsistent **CL**-consequence set is trivial.

[3] Provisionally, take this to mean that no contradiction can be derived from it by means of **CL**—see also Section 2.1.

[4] A logic is paraconsistent iff it does not assign the trivial consequence set to all inconsistent premise sets.

[5] The exceptions are the premise sets to which even **CLuN** assigns a trivial consequence set. Examples are the trivial premise set and the trivial premise set from which finitely many

case kind of adaptive logics, which are called flip-flop logics and will be discussed in later chapters, for example in Section 7.3. Here is an accurate description which, whoever, will become more transparent in later chapters. Flip-flops assign to a premise set Γ the consequence set of a certain Tarski logic if that set is non-trivial, and otherwise the consequence set of another Tarski logic, which is weaker than the former. In other words, a flip-flop logic behaves like one Tarski logic for some premise sets and like another Tarski logic for other premise sets. Adaptive logics that are not flip-flops proceed in a more refined way, as we shall see immediately, and are more fascinating than flip-flops. Still, some flip-flop logics have sensible applications, as we shall see in Chapter 3 and elsewhere.

Let us turn to the behaviour of adaptive logics that are not flip-flops. Many examples follow in subsequent chapters, including adaptive logics that handle inconsistent premises. Still, it is useful as well as easy to explain the difference in behaviour already at this point. Suppose that you consider some statements as certainties and others as expectancies. For the sake of an example, let $\neg r$ and $r \vee q$ (r-or-q) be your certainties and p and $\neg q$ your expectancies. An expectancy is typically a statement that you want to consider as true, unless your certainties prevent you from doing so. If your standard of deduction is **CL**, you consider q as certain because it is a **CL**-consequence of your certainties. This prevents you from considering the expectancy $\neg q$ as true. However, you will still consider the expectancy p as true, for p is not contradicted by your certainties. In Chapter 6 I define adaptive logics that do precisely this. These are not flip-flops. According to a flip-flop logic, no expectancy would be considered as true because a certainty, viz. q, prevents you from considering all expectancies as true. Adaptive logics that are not flip-flops will consider some expectancies as true even if not all of them can be taken to be true. In the present example, they ensure that the consequence set comprises as many expectancies as the certainties permit.

People clearly do not handle expectancies according to the idea underlying flip-flop logics. Suppose that you organize a party, expect sunny weather, and expect John to come to the party. If you find out that John is unable to come, you will not consider this a reason to stop expecting sunny weather.

Adaptive logics may be looked at in a different way. In a sense, a logic offers an 'interpretation' of a premise set. It determines what the premises 'mean' in settling what follows from them. In this respect, adaptive logics may be seen as specific combinations of two usual logics, a weaker one, which will be called the lower limit logic, and a stronger one, which will be called the upper limit logic.[6] The adaptive logic behaves exactly like the upper limit logic if the latter offers a sensible, viz. non-trivial, interpretation of the premise set Γ. If the upper limit logic does not assign a sensible interpretation to Γ, the adaptive logic still assigns to Γ at least all consequences that the lower limit logic assigns to it. For the logic called **L2** some paragraphs ago, the lower limit logic is **CLuN** whereas the upper limit logic is **CL**. Whenever the lower limit consequence set is non-trivial, interesting adaptive logics, viz. those that are not flip-flops, assign to Γ a non-trivial consequence set that is larger than that assigned by the lower limit. They interpret Γ in agreement with the upper limit *in as far as* this is possible.

formulas are removed.

[6]Some border cases are disregarded here. These will be cleared up in Chapter 5.

One way to phrase this is that adaptive logics validate certain inferences that are correct according to the upper limit, but not other such inferences. They validate those inferences that are justifiable on logical grounds.

By now, the reader should have an idea of what it means that a logic adapts itself to the premise set. Still, a misunderstanding might remain. The reader might think that adaptive logics depend on the insights of the person who applies the logic. If one applies an adaptive logic that handles certainties and expectancies to a complicated set of predicative certainties and expectancies, determining which expectancies are compatible with the certainties may be a difficult task. The situation may even be worse: for some premise sets, finding out whether the expectancies are compatible with the certainties may be too complex a task for a human being. Nevertheless, the adaptive logic assigns a specific consequence set to every premise set. And it should do so in order to be a logic as defined before. Whether humans are able to find out whether a formula belongs to the consequence set is immaterial.[7] Adaptive logics do not require any intervention on the part of the person applying them. Adaptive logics adapt *themselves* to the premises.

Some paragraphs ago, I referred to a 'standard of deduction'. Some take this to vary with the context, others consider it as absolute—the latter also use phrases like "the true logic" or "the canon of *a priori* reasoning". Adaptive logics are not candidates for the label "standard of deduction" if this is meant in an absolute sense. They are means to characterize, in a strictly formal way, forms of reasoning that were not hitherto recognized as formal.

As logicians heavily disagree on the standard of deduction, let me try to offer some comments on it. There are those who believe in the existence of an absolute, context independent, (but possibly unknown) standard of deduction. They distinguish between deductive forms of reasoning and other forms of reasoning. The former are defined as never leading from true premises to a false conclusion. Intuitionists, relevantists, and other brands of logicians will add some further properties, but these need not concern us here. Forms of reasoning that are not deductive are then called plausible, defeasible, argumentative, or by another name. One of the sensible stands on the matter is that the validity of forms of plausible reasoning depends on the world we live in, whereas the validity of deductive reasoning is independent of the world—what it depends on need not concern us here. A traditional example is that p is a deductive consequence of $p \wedge q$, whereas an inductive generalization is at best a plausible consequence of a set of data. This does not mean that the inductive reasoning leading from the data to the generalization is incorrect. All it means is that this inference is only plausible, and hence may be defeated, for example by new data.

According to some logicians all forms of reasoning depend on the world and are for this reason at best plausible. Only by experience and by theory building, they claim, we shall be able to find out which inferences are deductively correct. According to these logicians there is no absolute standard of deduction. Or rather, if there is one, it can only be known when our knowledge has become fully correct and stable, viz. at the proverbial end of time. Note that such a standard of deduction does not determine which reasoning forms are correct *for us*, that is in the historical period in which we live. We should reason in a way

[7]That humans are sometimes unable to find this out, involves complications for applying adaptive logics. These will be discussed in due time, especially in Chapter 10, but are unimportant in the present context.

that is suitable for our present knowledge, and this may be very different from the stable knowledge that might ideally be reached.

I obviously have a view on the positions considered in the previous paragraphs. I shall not take sides, however, because my view does not matter for the present book. Adaptive logics are neutral with respect to the discussed positions. They are compatible with all relevant (sensible) stands.

As was said before, if an absolute standard of deduction exists, adaptive logics are not candidates for it. I shall, however, write this book as if **CL** were the standard of deduction whereas adaptive logics lead only to plausible conclusions. This decision is inspired by pragmatic reasons. Of all usual logics, **CL** is most popular as well as easiest to handle.

A standard of deduction offers an easy way to classify adaptive logics in corrective ones and ampliative ones. The corrective ones deliver a consequence set that is weaker than the standard of deduction. They are typically applied to premise sets to which the standard of deduction assigns the trivial consequence set. In this sense they offer means to approach the standard of deduction as much as possible. Ampliative adaptive logics offer a richer consequence set than the standard of deduction, which explains their name. They are applied when, from a given premise set, one does not only want to derive the deductive consequences but also some defeasible or plausible consequences. Logicians that disagree on the standard of deduction will classify adaptive logics differently. My pragmatic classification agrees with the principled classification of classical logicians—they obviously dislike that my classification is pragmatic only.

It seems wise to add a word on the origin of adaptive logics. This does not lie in technical insights, but in an attempt to explicate reasoning processes that occur in actual reasoning, both everyday reasoning and scientific reasoning. As far as scientific reasoning is concerned, the processes are not located in contexts in which finished theories are formulated, but in contexts in which theories are forged or modified, and sometimes in contexts in which theories are applied. So adaptive logics are not intended to be used as the underlying logic of scientific theories,[8] but are intended for explicating problem-solving processes, especially creative processes or discovery processes, and for explicating reasoning processes related to the application of theories, for example to scientific explanation. In other words, adaptive logics are among other things a means to characterize methods in a precise way.

So the reasoning processes came first. Adaptive logics are an attempt to *explicate* the involved reasoning forms, which are numerous. Traditionally, the reasoning forms were either considered as incorrect or as too indistinct to allow for formal treatment.

The requirements on good explications, the *locus classicus* is obviously [Car50], RU: eisen vermelden? entail that adaptive logics have some unusual properties in comparison to more usual logics—the central cause of the unusual properties will be discussed in Section 1.2. Yet, adaptive logics can and should be presented in agreement with

[8]The underlying logic of a theory is the logic by which the theory is axiomatized. So, if Γ is the set of axioms of the theory and **L** is its underlying logic, the theory (on the statement view) can be identified with the consequence set that **L** assigns to Γ. This is the reason why such a theory is often denoted by the couple $\langle \Gamma, \mathbf{L} \rangle$. Where **L** is an adaptive logic, this consequence set may still be taken to be a (slightly unusual) theory—see the comments following Corollary 5.6.3. Occasionally, it may be useful to study such a theory as a (provisional) alternative for an existing one.

the strictest standards of rigour. Moreover, they can and should be studied, at the metalevel, by means of the marvellous set of tools that was developed by logicians of the past—a few new tools had to be developed, but they are very similar to what was given.

1.2 Dynamic Reasoning Forms

voetn.: + REFS

The reasoning forms explicated by adaptive logics are in a specific sense dynamic.[9] This is the central cause of the unusual properties of these logics. Actually, two forms of dynamics may occur: an external one and an internal one. The *external dynamics* consists in the fact that a conclusion may be withdrawn in view of *new information.* In other words, the reasoning process is non-monotonic: some conclusions derivable from a premise set may not be derivable any more if further premises are added. The *internal dynamics* occurs if a conclusion may be withdrawn in view of the *increased understanding of the premises.*

In both cases, a formerly drawn conclusion may be revoked. In the case of the *external* dynamics, this occurs because new information is gained while the reasoning is going on. In the case of the *internal* dynamics, the cause lies with the reasoning process itself: as it proceeds, the insight in the premises increases.[10] The external dynamics (non-monotonicity) is a property of the inference relation. The internal dynamics affects the actual reasoning, not the inference relation. The external dynamics results in an internal dynamics, but, as we shall see in this very section, the internal dynamics may occur in the absence of an external dynamics.

While the external dynamics is well-documented in the literature—see for example [Mak05] and its reference section—the internal dynamics is not. Nevertheless, it is a most familiar phenomenon. Everyone who takes the time to think about his or her convictions, will at some point decide them to comprise certain statements which he or she will revoke later, even if no new information was accepted in the meantime. The reason to revoke the statement may lie with good arguments against it, with better arguments for an alternative, or simply with the insight that the arguments in its favour are not sufficiently convincing. Similarly, all forms of ampliative reasoning applied in a scientific context (inductive generalization, abduction, analogy arguments, etc.) rely on a given system of knowledge and extend it. Every extension introduced by such a reasoning may later be revoked. It may turn out that the extension is incompatible with the original knowledge system, that a different extension is more elegant or more systematic, or that the arguments for choosing this specific extension are too weak.

It might seem that the dynamics can be avoided by introducing further requirements, for example that the extension is provably compatible with the knowledge system, that it has been demonstrated that there is no preferable alternative, etc. Introducing such requirements, however, will not only stop the dynamics but prevent one from reaching any conclusion at all. Compatibility

[9]The dynamics of a reasoning form should not be confused with dynamic aspects of the interpretation of statements. The latter are explicated by so-called dynamic logics.

[10]As we shall see in Section 4.10, this increasing insight does not depend on the person performing the reasoning, but is revealed by the steps of which the reasoning process consists.

is not a decidable relation: there is no algorithm for "A is compatible with Γ" even if Γ is a recursive set, which means that it is decidable whether a formula is a member of Γ. Even worse, there is no positive test for it; in other words, the set of formulas compatible with Γ is not even semi-recursive.[11] Other possible requirements lead to worse trouble.

There is a positive test for *in*compatibility. There is a mechanical procedure that, after finitely many steps, informs one that A is incompatible with Γ iff it is. The positive test for incompatibility, however, is not much use in the situation under consideration. If the positive test for incompatibility is successful, a candidate extension is ruled out, which is useful. But what one is interested in, in the given situation, are extensions that *are* compatible with the knowledge system. The point is not a theoretical one. We shall see in Section 2.1 that inconsistencies often occurred in the history of the sciences.

In the absence of a positive test, one has the choice between two alternatives: either one gives up reasoning towards a conclusion, or one opts for dynamic reasoning processes. It turns out that sensible humans are smart enough to take the latter option. The alternatives for reaching a decision are not very attractive: fortune-telling, throwing up a coin, blind faith, etc. So sensible humans *reason* towards conclusions and decide in view of them, even if they know that the conclusions may be overruled by new information or by a better understanding of present information. Proceeding thus has led to many mistakes, but also to great realizations, such as the sciences. Given this, it seems worthwhile to find a decent formal explication of dynamic reasoning processes.

Let us consider some examples of reasoning forms for which there is no positive test. The absence of a positive test is indeed the criterion for deciding that the consequence relation is characterized by an adaptive logic. The following list of examples is far from complete. Still, it seems useful to give the reader an idea of the large diversity of the involved forms of reasoning as well as of the frequency with which they occur.

We already came across handling inconsistency in Section 1.1. Consider the case in which a scientific (empirical or mathematical) theory T was meant to be consistent and was formulated with **CL** as its underlying logic, but turned out to be inconsistent. The historical literature—see Section 2.1 for references—is quite clear on such cases. Scientists do not dismiss T in order to start from scratch. They *reason from* T in search for a consistent replacement. They obviously do not reason in terms of **CL**, because that reduces T to triviality. They also do not reason in terms of some monotonic paraconsistent logic. As we shall see in Chapter 2, to do so would lead to a consequence set that is too weak. Scientists interpret T *as consistently as possible* and use this interpretation as a starting point for finding a consistent replacement for T.

Interpreting T as consistently as possible requires reasoning from T. This will be shown in Chapter 2, but it is not difficult to consider a simple example already here. Let T consist of $\neg p$, $p \vee r$, $\neg q \wedge p$, and $q \vee s$. As p follows from $\neg q \wedge p$, T requires p to behave inconsistently—let us say that both p and $\neg p$ are

[11]A *positive test* for a property of objects of a given kind is a mechanical procedure that leads after finitely many steps to the answer YES iff the object has the property. If the object does not have the property, the procedure may lead to the answer NO or may never halt. A property for which there is a positive test is also said to be semi-decidable or semi-recursive; the set of objects having the property is also said to be semi-decidable or semi-recursive. See [BJ89] or [BBJ02].

true *on* the theory. However, T does not require q to behave inconsistently: T requires $\neg q$ to be true, but does not require q to be true. Indeed, the only way to derive q from T is by explicitly or implicitly applying Ex Falso Quodlibet to p and $\neg p$.[12] Given that q behaves consistently on the theory, s is derivable from T: either q or s is true and q is false. Given that p behaves inconsistently, r is not derivable from $p \vee r$ and $\neg p$: either p or r is true and p is indeed true (together with $\neg p$). This approach requires a reasoning to settle whether a formula behaves consistently or inconsistently on the theory. In the absence of a positive test for consistency, the reasoning is necessarily dynamic.

Some readers may think that some ways to handle inconsistency avoid dynamics. For example, the Rescher–Manor consequence relations handle inconsistencies in a way that is very different from the approach suggested in the previous paragraphs—see [Res64, Res73, RM70]; see [BDP97, BDP99] for a survey and study of those consequence relations. Let us consider one of them, viz. the Weak Consequence relation. A is a *Weak Consequence* of Γ iff it is a **CL**-consequence of a consistent subset of Γ. Clearly, this definition does not involve any dynamics. However, as there is no positive test for consistency, the type of *reasoning* that leads to "A is a Weak Consequence of Γ" is necessarily dynamic. In Section 9.7 we shall see that every Rescher–Manor consequence relation is characterized by an adaptive logic.

Two side remarks are at hand. First, there are non-dynamic *definitions* for the consequence relations of adaptive logics. To be more precise, for every known adaptive logic **AL**, and for every (known or unknown) adaptive logic **AL** in the standard format from Chapter 4, it is possible to define A as an **AL**-consequence of Γ in a way that does not involve dynamics. The external dynamics does not prevent a non-dynamic definition of the consequence relation, and the internal dynamics does not even affect the consequence relation.[13] The second remark is that reasoning connected to the Weak Consequence relation displays the internal dynamics but not the external one. Indeed, the Weak Consequence relation is monotonic: A is a Weak Consequence of Γ iff it is a **CL**-consequence of a consistent subset of Γ and hence all Weak Consequences of Γ are weak consequences of $\Gamma \cup \Gamma'$ (for every Γ'). So the reasoning process leading to "A is a Weak Consequence of Γ" necessarily displays the internal dynamics, but the external dynamics is absent—some paragraphs ago, I promised to illustrate this.

We have seen that the absence of a positive test causes the internal dynamics. The internal dynamics is the distinctive trait of defeasible reasoning. A reasoning form is defeasible iff it necessarily contains conclusions that are drawn provisionally, in other words, conclusions that possibly need to be revoked later when more insight in the premises is gained. It is worth stressing this point be-

[12]Ex Falso Quodlibet (EFQ): to derive B from A and $\neg A$. Other names are Ex Contradictione Quotlibet and Explosion. One way to implicitly apply EFQ (splitting it up in two steps) proceeds by first applying Addition to obtain $p \vee q$ from p and next applying Disjunctive Syllogism to obtain q from $\neg p$ and $p \vee q$. Addition (Add): to derive $A \vee B$ from A (or from B). Disjunctive Syllogism: to derive B from $A \vee B$ and $\neg A$.

[13]Remember that a logic is a function. That the **L**-consequence relation holds between the premise set Γ and the formula A means that A belongs to the set assigned by **L** to Γ. The effect of the external dynamics is that the consequence set assigned to Γ need not be a subset of the consequence set assigned to $\Gamma \cup \Delta$. The internal dynamics concerns the way in which one may find out whether A belongs to the consequence set assigned to Γ, not the question whether A belongs to Γ.

cause the literature offers a somewhat misleading view on the matter. Defeasible reasoning is often identified with non-monotonic logics, but this is a mistake. All non-monotonic consequence relations are defeasible, but not the other way around. But there is more. All popular approaches to defeasible reasoning lack a proof theory. So they are unable to explicate defeasible reasoning. This is presumably the reason why the external and the internal dynamics are blurred.

For **CL** and most other common first-order logics **L** the situation is as follows. There is a positive test for "A is a **L**-consequence of a recursive premise set Γ", but not a negative test. Put differently, the set of **L**-consequences of a recursive Γ is semi-recursive but the set of formulas that are not **L**-consequences of Γ is not semi-recursive—if both were semi-recursive, both would be recursive. If one defines a concept in terms of such a logic, the concept may involve negative clauses: so-and-so is not derivable from the premises or so-and-so is not a **L**-theorem. Several examples from the literature follow, but the logic **L1** from Section 1.1 nicely illustrates the matter. A member A of Γ does not contradict itself just in case $\neg A$ is not a **CL**-theorem. But there is no positive test for **CL**-non-theoremhood. So there is no positive test for "B is a **L1**-consequence of Γ" even if Γ is recursive. Indeed, B is a **L1**-consequence of Γ iff there is an $A \in \Gamma$ such that $\neg A$ is not a **CL**-theorem and B is a **CL**-consequence of A. There is a positive test for "B is a **CL**-consequence of A", but there is no positive test for "$\neg A$ is not a **CL**-theorem". Summing up, even the most common logics allow for definitions that introduce consequence relations for which there is no positive test.

In the previous paragraph I stated that there is no positive test for "B is a **L1**-consequence of Γ" even if Γ is recursive. If Γ is not a recursive set, then the set of **L**-consequences of Γ is never semi-decidable, except for some border case logics **L**.[14] This is completely obvious and recognizing a logic as an exception is also an obvious task. For this reason I shall, in making claims about decidability or semi-decidability (and recursiveness and so on), not always mention the requirement that the involved sets are themselves decidable.

We have seen that the cause of the internal dynamics is the absence of a positive test for the consequence relation. Let us now consider some further examples from the literature. In each of them, a definition leads to a consequence relation for which there is no positive test.

A recent version of the theory of the process of explanation is presented by Ilpo Halonen and Jaakko Hintikka in [HH05]. In Section 6, they discuss the conditions on (nonstatistical) explanations (with a number of restrictions). The conditions (phrased in the notation of this book) concern an explanandum Pb, a background theory T (in which the predicate P occurs) and an initial condition (antecedent condition) I (in which b occurs). Among the six conditions are the following:

(iv) The explanandum is not implied by T alone.

(vi) I is compatible with T, i.e. the initial condition does not falsify the background theory.

There is no positive test for either of these conditions. In other words, no finite

[14]The exceptions comprise constant logics, which assign the same consequence set to every premise set. Some such logics play a role in this book, for example **Tr** and **Em**, but there are also logics that assign, for example, the consequence set $\{p\}$ to every premise set. There are also other exceptions, for example the logic that behaves just like **CL** whenever the premise set is recursive and assigns the empty set to every non-recursive premise set.

reasoning process can (in general) lead to the conclusion that Pb is explained by I and T.

So although the standard of deduction is **CL**, as appears from the listed conditions, the reasoning process that leads to the conclusion that I and T together explain Pb cannot possibly be explicated in terms of a **CL**-proof. The reasoning is *about* **CL**-derivability, and necessarily displays the internal dynamics. This is why it can only be explicated by a dynamic proof as described in Section 4.7.[15] It cannot be adequately handled in general, for example, by Hintikka's question logic as presented in [Hin99] and elsewhere.

ref in noot

Let us turn to an example from a very different domain: erotetic logic as developed by Andrzej Wiśniewski. According to [Wiś96] and [Wiś95], where, among other things, the evocation problem is studied and solved, a question Q is *evoked* by a set of declarative statements Γ iff the (prospective) presupposition[16] of Q is derivable from Γ but no direct answer of Q is derivable from Γ. Note that there is no positive test for non-derivability.[17] So, although the definition itself is unobjectionable, only a dynamic reasoning may (in general) lead to the conclusion that Q is evoked by Γ.[18]

The most common mechanism that, in specific situations, leads to general knowledge is inductive generalization. It has often been said that there is no logic of induction. This claim is ambiguous. If an absolute standard of deduction exists, it clearly does not validate inductive inferences. However, as we shall see in Chapter 3, there are (many) adaptive logics of inductive generalization. The underlying idea is clear and unambiguous. In the simplest case, one concludes to a generalization from a set of singular data. In minimally realistic cases, one has to take into account a variety of background knowledge, which moreover is defeasible in that the data might contradict it. This entails that several adaptive logics are invoked in contexts in which inductive generalizations are arrived at. All such inferences require that the generalization be compatible with the available knowledge that is not contradicted by the data. As there is no positive test for compatibility, the reasoning leading to inductive generalization is necessarily dynamic. Actually, the logic of inductive generalization is more refined than is suggested here. Still, the argument as presented here holds.

An example that we already met in Section 1.1 concerns logics handling expectancies. The absence of a positive test is striking: the retained expectancies need to be jointly compatible with the certainties. Moreover, expectancies come in degrees. Some statements are strongly expected to be true, others a bit less strongly, etc. So one has first to add the strongest jointly compatible expectancies to the certainties. To the result of this one has to add the next strongest jointly compatible expectancies, and so on. A positive test is lacking for every step. So each step requires dynamic reasoning.

This raises an interesting problem concerning the combination of dynamic

---

[15]See [BM01a] and Section XXX for adaptive logics that explicate the forms of reasoning involved in the search for explanations. A very different approach to nearly the same problem is presented in [MB06], but the explicated reasoning is just as dynamic.

[16]The prospective presupposition, for example, of a whether-question is the disjunction of its direct answers. Thus the prospective presupposition of "Did Mary or John or Joan come?" is 'Mary came or John came or Joan came." Strictly speaking, a question has many prospective presuppositions because a statement equivalent to an answer also counts as an answer.

[17]I mean **CL**-non-derivability, in agreement with the cited papers, but the matter is the same for any logic that can sensibly be applied in this context.

[18]See [Meh01] for the adaptive logic that explicates the dynamic reasoning.

reasoning forms. In the example from the previous paragraph, one might think that even the first step (adding the strongest expectancies to the certainties) might require an infinite time, whence no human can possibly ever move on to the second step (adding the next strongest expectancies to the result of the first step). However, humans clearly proceed differently. They are able to revise even their judgement on the strongest expectancies after weaker expectancies have been invoked. We shall see in Section 6.2.2 that adaptive logics explicate precisely this way of proceeding.

Let me mention a final set of examples: all non-monotonic logics. The underlying idea of reasoning towards a non-monotonic conclusion always comprises two elements. Some formula should be derivable from the premises by a given monotonic logic **L** and some other formula should not be derivable from the premises by **L**.[19] So if **L** is a usual logic of (at least) the same complexity as **CL**, there is no positive test for non-derivability in the predicative case.

What precedes is by no means an exhaustive list of the reasoning mechanisms, or even of the types of reasoning mechanisms, for which a positive test is lacking. My aim was not to present such a list. The only point I tried to make was this: an internal dynamics is present in certain very common forms of human reasoning that are important for understanding the way in which humans arrive at knowledge and revise it.

1.3 Specific Features

Adaptive logics have a number of specific properties that distinguish them from many other approaches to defeasible reasoning. By the very fact that they are logics, they present a *qualitative* approach to defeasible reasoning. Take inductive generalization as an example. Most approaches to inductive generalization rely essentially on probabilities of some sort or other. One of the difficulties of these approaches lies in the manifold of 'measure functions' or 'priors'. Some of these lead to closely similar results, others to wildly different results. As the history of Carnapian inductive logic illustrates, no measure function has been shown to be adequate in general—Carnap's plan failed in this sense—and while the choice of certain measure functions can be justified with respect to some extremely simplistic applications, this justification applies just as well to an infinity of measure functions. Moreover, the required construction and the connected calculations are terribly complex and none of all this is even remotely related to actual inductive reasoning.

Adaptive logics proceed along a different road. They assign to every set of data, interpreted in terms of background knowledge in realistic cases, a specific consequence set, viz. a set of generalizations. I do not claim that all problems are solved. Nor do I claim that an explication in terms of logical or personal probabilities is worthless. Nevertheless, to the extent that adaptive logics of inductive generalization are adequate, the burden of proof is on those who claim probabilistic approaches to be superior.

A second feature is that adaptive logics are formal logics. In other words, whether a formula is a consequence of a premise set according to an adaptive logic depends only on the logical form of the premises and of the conclusion. All

[19]Obviously, the matter may be formulated in semantic terms, but that does not reduce the complexity of the reasoning.

common logics are also formal. To be more precise they fulfil a certain formal criterion which is called Uniformity. As we shall see in Section 5.5, adaptive logics are not uniform.[20] Actually, no logic that explicates defeasible reasoning forms can be uniform. Nevertheless, adaptive logics are formal logics because they fulfil another formal criterion, which I shall call Punctual Uniformity. Common logics also fulfil this criterion. In this sense adaptive logics stay maximally close to the standard approach that is used by logicians. Some examples of logics that are not punctually uniform will be presented in Section 5.5.

The third specific trait of adaptive logics is that they have *proofs*. This may be partly a lucky historical accident; the first adaptive logic grew out of considerations concerning proofs. The proofs are dynamic, but the rules governing them are simple and the matter was thoroughly studied. Most other approaches proceed either in terms of models, or 'extensions', or tableaus, but fail to offer proofs—some people are even convinced that the logics that are popular (and often developed) within artificial intelligence concern inference relations that cannot be characterized in terms of proofs. We shall see in Chapter 9 that this is a mistake.

The main advantage of an approach in terms of proofs is that it is much closer to actual defeasible reasoning. Consider abduction as an example. When I wake up and the plants in my garden are sopping wet, I conclude that it has rained during the night. This is obviously a defeasible consequence. Some funny people may have wetted my garden while I was soundly asleep and one might think up a couple of other outlandish explanations. Nevertheless, the actual reasoning reduces to a one step inference. It is so immediate that some people will even (erroneously) claim to see that it has rained.

Consider an explication of the abduction "it has rained" in terms of models. This requires reasoning about an infinite multitude of infinite entities. Sometimes one is able to reduce the matter to finitely many finite entities, such as the extensions of default 'logics'. However, this reduction is often realized by arbitrary decisions. Moreover, the result still requires introducing entities, and often complex ones, that are constructed and selected by complex forms of reasoning. Neither the entities nor the required reasoning occurs anywhere in the actual abductive reasoning. This holds also for approaches in terms of tableaus—just think about the multiplicity of tableaus, generated by 'splitting', that have to be considered in the case of minimally realistic examples.

If one explicates the abduction in terms of an adaptive proof, the result is a one step inference. So, again, to the extent that adaptive explication is adequate, the burden of the proof is on those who claim that it is superior to proceed in terms of more complex entities and more complex reasoning forms.

Let me avoid a misunderstanding at this point. The involved reasoning processes are complex in that the underlying inference relation is complex. Defeasible reasoning is way more complex than deductive inference. So it is unavoidable that the inference relation characterized by an adaptive logic is complex. This complexity, however, does not derive from the proofs themselves. Indeed, the proofs are governed by simple rules and a simple definition. The complexity concerns the question whether a conclusion that has been drawn at a certain point is final or not. This matter will be discussed extensively in Chapter 10.

[20]For the time being, just think about the Uniform Substitution rule that can be applied to inferential statements $A_1, \ldots, A_n \vdash B$ in usual propositional logics.

Of course, adaptive logics have a semantics. So the reasoning may be shown correct in model theoretic terms. And tableau methods have been devised for adaptive logics. They display a certain interest because they offer criteria, in certain cases, for settling whether a conclusion is final or not. Devising a semantics and tableaus is essential for studying the logics according to the standards of the profession. Nevertheless, the dynamic proofs of adaptive logics form the explication for actual defeasible reasoning.

Many other specific features of adaptive logics will be highlighted in this book, for example the easy identification of equivalent premise sets discussed in Section 5.7. However, the three points mentioned in this section deserved to be stated from the very outset.

1.4 Languages

Although most conventions in this book will be clear from the context, those that concern formal languages and their elements are better spelled out explicitly. You may skip the rest of this section and return here, or consult the index to be referred here, when you need the conventions introduced below.

Throughout this book, $\mathcal{L}$ is used as a *variable* for languages or language schemata,[21] $\mathcal{F}$ is the set of formulas of $\mathcal{L}$ and $\mathcal{W}$ the set of closed formulas[22] of $\mathcal{L}$—closed formulas are those that contain no free variables. All language schemata will be taken to have a denumerable alphabet (the set of symbols occurring in formulas of the language schema is denumerable). All formation rules are as usual and formulas are strings of finitely many symbols.

The standard predicative language schema is denoted by $\mathcal{L}_s$. The logical symbols of $\mathcal{L}_s$ will be $\supset$, $\wedge$, $\vee$, $\equiv$, $\neg$, $\forall$, $\exists$, and $=$. The sets of schematic letters of $\mathcal{L}_s$ will be named as follows: $\mathcal{S}$ is the set of sentential letters, $\mathcal{C}$ the set of individual constants, $\mathcal{V}$ the set of individual variables, and $\mathcal{P}^r$ the set of predicative letters of rank r for every natural number $r > 0$.

The standard modal language schema $\mathcal{L}_m$ is obtained by extending $\mathcal{L}_s$ with the modal operators $\Box$ and $\Diamond$.

Several logics weaker than **CL** will be considered. Such logics assign to some logical symbols a meaning that is different from that assigned by **CL**. When this is the case, it is useful for our purposes to extend the language schema with new logical symbols that have the same meaning as those of **CL**. These symbols will not occur in the premises or conclusion. In other words, standard applications will be taken to be phrased within the standard language schema. The symbols are merely introduced to simplify the logician's work.[23] The new symbols, which will be called the *classical symbols*, will be written as $\check{\supset}$, $\check{\wedge}$, $\check{\vee}$, $\check{\equiv}$, $\check{\neg}$, $\check{\forall}$, $\check{\exists}$, and $\check{=}$, occasionally also $\check{\Box}$ and $\check{\Diamond}$. Extending $\mathcal{L}_s$ with the classical counterparts of its symbols will result in the language schema $\mathcal{L}_S$; doing the same to $\mathcal{L}_m$ will result in the language schema $\mathcal{L}_M$. nazien wat nodig

[21] A language schema may be seen as the common structure of a set of languages. It is built up from logical symbols and schematic letters—see below in the text. If the schematic letters are replaced by constants (for sentences, predicates, ...), one obtains a formal language.

[22] In all language *schemata* I shall consider, only closed formulas will be well-formed, whence $\mathcal{W}$ will refer to closed formulas.

[23] The precise way in which they are introduced is discussed in Section 4.3. As we shall see, introducing the symbols is 'harmless'. Moreover, the symbols simplify the functioning of the logics at the level of the object language as well as the metalinguistic proofs.

Sometimes a classical symbol may be defined from standard symbols. Even in those cases, it is often easier to introduce the classical symbol directly, by rules for the proof theory and by clauses (and similar means) for the semantics. I shall not use a classical symbol in a context in which the corresponding standard symbol has the same meaning. Nevertheless, I shall always consider the classical symbols to belong to (and to be defined suitably in) the extended language schema $\mathcal{L}_S$ or $\mathcal{L}_M$.

Occasionally, I shall introduce another language schema or a language. These will always be introduced explicitly in the context in which they are required.

In order to keep the description of the semantic systems as simple as possible, I introduce pseudo-language schemata. Every pseudo-language schema will always be denoted by $\mathcal{L}_\mathcal{O}$, but it will always be clear from the context what is meant. Let me briefly expand on this here.

The domain of a model is a set. So the maximal cardinality of domains depends on your personal set theory: it is the cardinality of the largest set. According to most logician's preferences, the largest set is non-denumerable. At the same time, these logicians agree that languages and language schemata are supposed to have at most a denumerable set of symbols. So, for some models M, not all elements of the domain of M can be named by a linguistic entity.

By a pseudo-language schema I mean a language schema extended with a set $\mathcal{O}$ of pseudo-constants, where the cardinality of $\mathcal{O}$ is the cardinality of the largest set. Given that the cardinality of $\mathcal{O}$ is that of the largest set, $\mathcal{O}$ comprises a sufficient supply of names to name every element of the largest domain. The presence of the pseudo-language schema will render the wording simpler and will make it easier to grasp what is meant.

Of course, I shall take care that anything phrased in $\mathcal{L}_\mathcal{O}$ can sensibly be 'translated' into statements about the domain of a model. If we can sensibly talk about non-denumerable domains, we can also sensibly talk *about* pseudo-language schemata. This is precisely what I shall do in the semantic systems—a first example follows in Section 1.7.

As said before, $\mathcal{L}_\mathcal{O}$ will always refer to a specific pseudo-language schema and only the context will determine the pseudo-language schema. Considering a semantics for a language schema $\mathcal{L}$, $\mathcal{L}_\mathcal{O}$ will refer to the pseudo-language schema obtained by extending $\mathcal{L}$ with the set of pseudo-constants $\mathcal{O}$.

At some points I shall need a language schema $\mathcal{L}$ extended with a *denumerable* set of new constants, which will be taken from $\mathcal{O}$. The resulting language schema will be called $\mathcal{L}_o$. The precise language schema denoted by $\mathcal{L}_o$ will be specified if it is not clear from the context.

The letter $\mathcal{F}$ with a subscript attached to it will denote the formulas of the language schema denoted by a letter $\mathcal{L}$ with the same subscript; the letter $\mathcal{W}$ with a subscript attached to it will denote the closed formulas of the language schema denoted by a letter $\mathcal{L}$ with the same subscript. Thus $\mathcal{F}_s$ denotes the formulas of the language schema $\mathcal{L}_s$ and $\mathcal{W}_o$ denotes the closed formulas of the language schema $\mathcal{L}_o$.

In the *metalanguage*, I use the following metavariables, possibly with subscripts or superscripts:

zien wat nodig – ρ niet gebruikt

metavariables	for schematic letters for
α, β, γ, ...	individual variables and constants
π, ρ, ...	predicates
A, B, C, ...	(open and closed) formulas

Expressions such as π^r indicate that the rank of predicate π is r. Where $\alpha \in \mathcal{V}$ and $\beta \in \mathcal{C} \cup \mathcal{O}$, $A(\alpha)$ is a formula in which α is free and $A(\beta)$ is the result obtained by replacing every free occurrence of α in $A(\alpha)$ by β.

Occasionally, I shall need the *metametalanguage* to talk about metalinguistic expressions. ${}^m\mathcal{W}$ will denote the set of well-formed metalinguistic formulas, obtained, for example, from object language formulas by replacing every schematic letter by a suitable metavariable. I shall use the following metametalinguistic variables: A, B, ... as variables for metalinguistic formulas, P^r as a variable for metavariables for predicates of rank r, a, b, ... as variables for metavariables for individual constants and individual pseudo-constants, and x, y, ... as a variable for metavariables for individual variables.

1.5 Static Proofs

We have seen that a logic is a function $\mathbf{L}\colon \wp(\mathcal{W}) \to \wp(\mathcal{W})$. So in standard functional phraseology the $\mathbf{L}$-consequence set of Γ would be denoted by $\mathbf{L}(\Gamma)$ and that A is $\mathbf{L}$-consequence of Γ would be expressed by $A \in \mathbf{L}(\Gamma)$. The usual logical phraseology is different. Logics are characterized by a proof theory and a semantics, sometimes also by other means. For every such characterization, there are standard ways to refer to the consequence relation.

The $\mathbf{L}$-consequence relation as fixed by 'the' proof theory is denoted by $\vdash_{\mathbf{L}}$ and is often called the derivability relation. The expression $\Gamma \vdash_{\mathbf{L}} A$ denotes that A is $\mathbf{L}$-derivable from the premise set Γ. That A is not $\mathbf{L}$-derivable from the premise set Γ is written as $\Gamma \nvdash_{\mathbf{L}} A$.

One often introduces $Cn^{\mathcal{L}}_{\mathbf{L}}(\Gamma) =_{df} \{A \mid \Gamma \vdash_{\mathbf{L}} A; A \in \mathcal{W}\}$, the (syntactic) $\mathbf{L}$-consequence set of Γ in the language $\mathcal{L}$. Where the context disambiguates the matter, I shall sometimes drop the references to the language or to the logic.

That A is a *theorem* of $\mathbf{L}$, usually written as $\vdash_{\mathbf{L}} A$, may be defined in two ways: (i) $\emptyset \vdash_{\mathbf{L}} A$ or (ii) $\Gamma \vdash_{\mathbf{L}} A$ for all Γ. For monotonic logics—see below—both definitions are provably coextensive; for non-monotonic logics they are not. The logics we consider in this section have static proofs and such logics are monotonic. So both definitions will do. Let us now have a closer look at the proof theory.

In a book on adaptive logics, the proof theory requires special attention. While usual logics have static proofs, adaptive logics have dynamic proofs—these will be introduced in Section 4.7. As such proofs have received nearly no attention in the literature, they deserve to be spelled out in an explicit way. This has two effects for the present book. Logicians have shown relatively little interest in proofs. They often merely quote Hilbert's definition, which actually is the semi-accidental result of the failed enterprise called formalism. Apart from that, logicians tend to follow a Tarskian approach, characterizing consequence relations in terms of their properties (for example reflexivity) rather than in terms of the proofs to which they refer. In this respect, the present book is an exception. The second effect is related to the fact that I want to show that dynamic proofs are simple and natural entities. Dynamic proofs are most easily described in terms of annotated proofs (proofs with a number and a justification on each line). In order to simplify the comparison, I shall describe static proofs in terms of annotated proofs as well.[24] Moreover, I shall spell out static proofs in a rather meticulous way. Finally, I shall not only pay attention to static proofs as *results*, but also to the *process* by which they are generated.

The central elements of annotated proofs are rules, lines, and lists of lines.[25] These elements cannot be defined in such a way that they are completely independent of each other.

A *line* of a static annotated proof will be a triple comprising a line number, a formula, and a justification. 'Number' may be taken in the broad sense here: apart from natural numbers, expressions such as 5.4 or 5a will be considered acceptable line numbers.[26] The *justification* of a line l is a couple $\langle N_l, R_l \rangle$ in which N_l is a (possibly empty) set of lines (referred to by their numbers) and plaats R_l is a S-rule as introduced in the next paragraph.

The rules that lead to static proofs will be called S-rules.[27] A *S-rule* is a metalinguistic expression of the form Υ/A, read as "to derive A from Υ", in which A is a metalinguistic formula and Υ is a recursive set of metalinguistic formulas. A S-rule specifies that from formulas of a certain form another formula of a corresponding form may be derived. I shall say that A is *the result* of applying the rule Υ/A to Υ. A S-rule is *finitary* iff Υ is a finite set. The members of Υ will be called the *local premises*. Note that this is a technical term, which I use for lack of a better one. They should obviously not be confused with the premises of the proof.

S-rules may have a *restriction* attached to them. If this is the case, it is essential that it can be decided whether the restriction is fulfilled by inspecting the list of lines to which the application of the rule belongs. Examples of such restrictions occur, for example, in **CL** as described in Section 1.7. The only restriction worth commenting upon is the one specifying that a rule Υ/A may only be applied if the members of Υ are theorems (of the logic). Whether the restriction is fulfilled may be seen from (and may be defined in terms of) the *path* of the members of Υ in the list of lines. Where L is a list of lines, the *path* of line l of L is the smallest set Σ fulfilling (i) $l \in \Sigma$ and (ii) if $l' \in \Sigma$, then $N_{l'} \subseteq \Sigma$—$N_{l'}$ was defined in the next to last paragraph. Where A is the formula of line l, the path of l is also called the path of (this occurrence of) A.[28]

If any formula is to be derivable from any premise set, one of the S-rules needs to have the form $\emptyset/\mathsf{A}$, possibly with a restriction attached to it. If there is no restriction, A is usually called an axiom schema.[29] Some logics have a set

[24] Another reason to consider annotated proofs is that non-annotated proofs are parasitic on them—see the last paragraph of this section.

[25] A list is an enumeration of a set. In the list, each member of the set is associated with a positive integer, which indicates the place of the member in the list—see [BBJ02, Ch. 1].

[26] The line numbers will be supposed to be in some alphabetical order in the stages defined below. This is obviously a conventional matter. All that matters is that each line has a 'number' that identifies it in a unique way. This allows one to refer unambiguously to a line in the justification of another line.

[27] The name S-rule refers to the fact that these rules are typical for logics that have static proofs; it distinguishes them from the more general rules introduced in Section 4.7.

[28] If the premise rule—see below in the text—has not been applied in the path of (a specific occurrence of) A in the proof, then A is a theorem. That the converse does not hold is no problem. Every theorem may be derived in such a way that no premise occurs in its path.

[29] So axiom schemas are actually rules—axioms are formulas of the object language. Explicit definitions may also be seen as (couples of) rules. The definition $\mathsf{A} =_{df} \mathsf{B}$ corresponds to the

of axioms. For these it is sensible to have an Axiom rule: "If A is an axiom, then $\emptyset/A$." A decidable set of axiom schemas is obviously a means to define a decidable set of axioms. The most popular restricted S-rule of the form $\emptyset/\mathsf{A}$ is the premise rule Prem. Where Γ is the set of premises, Prem reads: "If $A \in \Gamma$, then $\emptyset/A$."

There seem not to be many alternatives for Prem. If the logic has no rule of the form $\emptyset/\mathsf{A}$, then it is the empty logic **Em** for which $Cn_{\mathbf{Em}}(\Gamma) = \emptyset$ for all Γ.[30] If the only rule of that form is unrestricted, **L** is the constant logic **Tr** according to which $\Gamma \vdash A$ for all Γ and A.[31] If rules of the form $\emptyset/\mathsf{A}$ introduce only axioms (of the logic), the resulting logic is a constant function that assigns the same consequence set to all premise sets, viz. the set of theorems of the logic. If rules of that form introduce only formulas of a specific logical form, as does $\emptyset/A \vee \neg A$, then the resulting logic is again the constant function that assigns to all premise sets the set of theorems of this logic. The only remaining alternative is that the rules of the form $\emptyset/\mathsf{A}$ introduce formulas that are functions of the premises, for example "If $A \in \Gamma$, then $\emptyset/\neg A$." or "If $A, B \in \Gamma$ and $A \neq B$, then $\emptyset/A \vee B$." or "If $A \in \Gamma$, then $\emptyset/A \wedge (B \vee \neg B)$." All such rules lead to logics that assign to a premise set Γ the consequence set that a different and more 'natural' logic assigns to a function of Γ. It is not clear that such rules would be justifiable in any context or would have sensible applications. As the only other logics that lack Prem are uninteresting and exceptional border cases, viz. **Em** and logics for which $Cn_{\mathbf{L}}(\Gamma) = Cn_{\mathbf{L}}(\emptyset)$ for all Γ, I now stipulate that every set of S-rules contains Prem—put more elegantly, I shall disregard other sets of S-rules in this book.

In the sequel of this section, $\mathcal{R}$ will always denote a set of S-rules (which contains Prem) and "line" will always mean a line of a static annotated proof. Given a set $\mathcal{R}$ of S-rules and a list L of lines, a line l of L is $\mathcal{R}$-*correct* iff (i) all members of N_l precede l in the list, (ii) $R_l \in \mathcal{R}$, and (iii) the formula of l is obtained by application of R_l to the formulas of the lines N_l.

We are now in a position to move on to the definition of static proofs, which will proceed in terms of stages. I first introduce five definitions that refer to a set of S-rules $\mathcal{R}$.

Definition 1.5.1 *A* $\mathcal{R}$-stage *from (the premise set)* Γ *is a list of* $\mathcal{R}$*-correct lines.*

Definition 1.5.2 *Where* L *and* L' *are* $\mathcal{R}$*-stages from* Γ*,* L' *is an* extension *of* L *iff all elements that occur in* L *occur in the same order in* L'*.*

Definition 1.5.3 *A static* $\mathcal{R}$-proof from Γ *is a chain*[32] *of* $\mathcal{R}$*-stages from* Γ*, the first element of which is the empty list and all other elements of which are extensions of their predecessors.*

Definition 1.5.4 *A static* $\mathcal{R}$-proof of A from Γ *is a static* $\mathcal{R}$*-proof from* Γ *in which, from a certain stage on, there is a line that has* A *as its formula.*

S-rule: from a formula A that contains an occurrence of A, to infer the formula obtained from A by replacing A by B, and *vice versa*.

[30] **Em** should not be confused with the zero logic $\mathbf{CL}\emptyset$ defined in Section 8.4.

[31] The name refers to the fact that this logic deserves to be called trivial. It should not be confused with the modal logic **Triv**. Adding Prem to **Tr** results in **Tr**.

[32] I am indebted to Andrzej Wiśniewski for the idea to see a (dynamic) proof as a chain of stages.

In view of Definition 1.5.3, Definition 1.5.4 comes to: a static $\mathcal{R}$*-proof of* A *from* Γ is a static $\mathcal{R}$-proof from Γ in which A is the formula of a line of a stage.

Definition 1.5.5 $\Gamma \vdash_{\mathcal{R}} A$ (A is $\mathcal{R}$-derivable from Γ) *iff there is a static* $\mathcal{R}$*-proof of* A *from* Γ.

The five preceding definitions enable us to delineate a specific set of logics, the members of which will turn out to have some interesting and unexpected properties.

Definition 1.5.6 *A logic* **L** has static proofs *iff there is a recursive set* $\mathcal{R}$ *of S-rules such that* $\Gamma \vdash_{\mathbf{L}} A$ *iff* $\Gamma \vdash_{\mathcal{R}} A$.

The name "static proof" refers to the fact that every formula that is derived at some stage remains derived at all subsequent stages. We shall see in Section 4.7 that this is different for dynamic proofs. All adaptive logics in standard format have dynamic proofs. Note that some logics, for example second order logic, have not even dynamic proofs. For now, let us continue to concentrate on logics that have static proofs.

Nearly every rule Υ/A has applications to sets of formulas with a lower cardinality than that of Υ and so generates a recursive set of "more specific rules". The rule $A/A \wedge A$ is in this sense more specific than $A, B/A \wedge B$—note that I do not write braces around the members of Υ when I enumerate them. In the same way, the infinitary rule $A, C_1 \wedge D_1, C_2 \wedge D_2, \ldots /A \vee B$ generates the more specific finitary rule $A, C_1 \wedge D_1/A \vee B$—the presence of a single formula of the form $C_1 \wedge D_1$ is sufficient to warrant the presence of a formula of the form $C_i \wedge D_i$ for all i. In general, every infinitary S-rule R generates zero or more finitary rules. The set of these rules, say $\text{fin}(R)$, is recursive.

Theorem 1.5.1 *If* $\mathcal{R}$ *is a recursive set of S-rules, then there is a recursive set* $\mathcal{R}'$ *of finitary S-rules such that* $\Gamma \vdash_{\mathcal{R}'} A$ *iff* $\Gamma \vdash_{\mathcal{R}} A$.

Proof. $\mathcal{R}$ is a recursive set of S-rules. Let $\mathcal{R}''$ be the result of replacing in $\mathcal{R}$ every infinitary rule by the recursive set $\text{fin}(R)$. Obviously $\mathcal{R}''$ is a recursive set.

Consider a $\mathcal{R}$-proof. In view of Definitions 1.5.1–1.5.3, the proof is a chain of lists of $\mathcal{R}$-correct lines. Suppose that one of the lists contains an application of an infinitary rule Υ/A. The result of the application occurs at a finite place n in the list. So the application can only rely on finitely many formulas that occur on lines preceding line n. It follows that the application can be justified by one of the rules in $\text{fin}(\Upsilon/\mathsf{A})$. ■

Corollary 1.5.1 *A logic* **L** has static proofs *iff there is a recursive set* $\mathcal{R}$ *of finitary S-rules such that* $\Gamma \vdash_{\mathbf{L}} A$ *iff* $\Gamma \vdash_{\mathcal{R}} A$.

It goes without saying that different sets of rules may define the same logic. Whenever **L** has static proofs, let $\mathcal{R}_{\mathbf{L}}$ be a recursive set of finitary S-rules such that $\Gamma \vdash_{\mathbf{L}} A$ iff $\Gamma \vdash_{\mathcal{R}_{\mathbf{L}}} A$. There is such a set in view of the last corollary.

It is easily provable that all logics that have static proofs share a set of interesting properties. The remaining part of this section will be devoted to those properties.

Definition 1.5.7 *A* standard $\mathcal{R}_{\mathbf{L}}$*-proof of* A *from* Γ *is a* $\mathcal{R}_{\mathbf{L}}$*-proof of* A *from* Γ *in which* A *is the formula of the* last *line of the* last *stage.*

Theorem 1.5.2 *If* **L** *has static proofs, then* $\Gamma \vdash_{\mathbf{L}} A$ *iff there is a standard* $\mathcal{R}_{\mathbf{L}}$*-proof of* A *from* Γ.

Proof. Suppose that the antecedent is true. In view of Definition 1.5.4, there is a $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ in which A is the formula of a line of a stage L on. Let l be such a line of stage L and let L' be the result of truncating L after line l. L' is the last stage of a $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ. ■

The usual definition of a (static) proof of A from Γ identifies a $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ with the last stage of a standard $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ. So, if **L** has static proofs, $\Gamma \vdash_{\mathbf{L}} A$ holds according to the usual definition just in case it holds according to the definitions of the present section.

Theorem 1.5.3 *If* **L** *has static proofs, then* **L** *is Compact*[33] *(if* $A \in Cn_{\mathbf{L}}(\Gamma)$ *then* $A \in Cn_{\mathbf{L}}(\Gamma')$ *for some finite* $\Gamma' \subseteq \Gamma$*).*

Proof. Suppose that the antecedent is true and that $\Gamma \vdash_{\mathbf{L}} A$. In view of Theorem 1.5.2, there is a standard $\mathcal{R}_{\mathbf{L}}$-proof p of A from Γ. Let L be the last stage of p and let Γ' comprise the members of Γ that have been introduced in L as premises. $\Gamma' \subseteq \Gamma$ is a finite set and p is a $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ'. ■

Theorem 1.5.4 *If* **L** *has static proofs, then* **L** *is Reflexive (*$\Gamma \subseteq Cn_{\mathbf{L}}(\Gamma)$*).*

Proof. Suppose that the antecedent is true and that $A \in \Gamma$. Let L be the list comprising a single line that has A as its formula and "Premise" as its justification. By Definition 1.5.4, the chain that has L as its sole stage is a static $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ. So $A \in Cn_{\mathbf{L}}(\Gamma)$ by Definitions 1.5.5 and 1.5.6. ■

Theorem 1.5.5 *If* **L** *has static proofs, then* **L** *is Transitive (if* $\Delta \subseteq Cn_{\mathbf{L}}(\Gamma)$*, then* $Cn_{\mathbf{L}}(\Delta) \subseteq Cn_{\mathbf{L}}(\Gamma)$*).*

Proof. Suppose that the antecedent is true, that $A \in Cn_{\mathbf{L}}(\Delta)$, and that $\Delta \subseteq Cn_{\mathbf{L}}(\Gamma)$. In view of Theorem 1.5.3, there is a finite $\Delta' \subseteq \Delta$ such that $A \in Cn_{\mathbf{L}}(\Delta')$ and $\Delta' \subseteq Cn_{\mathbf{L}}(\Gamma)$. Let $\Delta' = \{B_1, \ldots, B_n\}$. In view of Definitions 1.5.5, 1.5.6 and 1.5.3, there is, for each B_i $(1 \leq i \leq n)$, a standard $\mathcal{R}_{\mathbf{L}}$-proof p_i of B_i from Γ and there is a standard $\mathcal{R}_{\mathbf{L}}$-proof p_{n+1} of A from Δ'. Let L_i be the last stage of p_i and let L_{n+1} be the result of deleting from the last stage of p_{n+1} all lines on which a member of Δ' is introduced by the premise rule. The list obtained by concatenating $L_1, \ldots, L_n, L_{n+1}$ and adjusting the line numbers[34] is easily seen to be a correct $\mathcal{R}_{\mathbf{L}}$-stage from Γ and is a stage of a static $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ. So $A \in Cn_{\mathbf{L}}(\Gamma)$ by Definitions 1.5.5 and 1.5.6. ■

The preceding proof requires a replacement of some line numbers and of the references to them. The matter is actually obvious, as may be seen from the following example (in terms of **CL**-proofs).

1	$p \wedge (p \supset q)$	Premise
2	p	1; Simplication

[33] Standard properties, such as Compactness, will not be introduced in separate definitions but in the theorems in which they first occur, for example Theorem 1.5.3 for Compactness.

[34] For example, line number j in L_1 may be replaced by $1.j$, etc. In the justifications of L_{n+1}, a reference to the number of a deleted line on which B_j is introduced as a premise is replaced by the new number of the last line of L_j.

1	$p \wedge (p \supset q)$	Premise
2	$p \supset q$	1; Simplication

1	p	Premise
2	$p \supset q$	Premise
3	q	1, 2; MP

The concatenation of the three proofs looks as follows—obviously one may delete line 3 and refer to 1 instead of 3 in the justification of line 4. Note the adjustment of the justification of line 5.

1	$p \wedge (p \supset q)$	Premise
2	p	1; Simplication
3	$p \wedge (p \supset q)$	Premise
4	$p \supset q$	3; Simplication
5	q	2, 4; MP

If **L** is Transitive, then, as $Cn_{\mathbf{L}}(\Gamma) \subseteq Cn_{\mathbf{L}}(\Gamma)$ holds for all logics, $Cn_{\mathbf{L}}(Cn_{\mathbf{L}}(\Gamma)) \subseteq Cn_{\mathbf{L}}(\Gamma)$. If **L** is Reflexive, $\Gamma \subseteq Cn_{\mathbf{L}}(\Gamma)$. These give us the following lemma and corollary.

Lemma 1.5.1 *If* **L** *is Reflexive and Transitive, then* **L** *has the Fixed Point property* ($Cn_{\mathbf{L}}(Cn_{\mathbf{L}}(\Gamma)) = Cn_{\mathbf{L}}(\Gamma)$)*.*

Corollary 1.5.2 *If* **L** *has static proofs, then* **L** *has the Fixed Point property.*

The Fixed Point property is also called Idempotence. If **L** has the Fixed Point property, one also says that $Cn_{\mathbf{L}}(\Gamma)$ is a fixed point.

Theorem 1.5.6 *If* **L** *has static proofs, then* **L** *is Monotonic* ($Cn_{\mathbf{L}}(\Gamma) \subseteq Cn_{\mathbf{L}}(\Gamma \cup \Gamma')$ *for all* Γ')*.*

Proof. In view of Definitions 1.5.1–1.5.4, every $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ is a $\mathcal{R}_{\mathbf{L}}$-proof of A from $\Gamma \cup \Gamma'$. So the theorem follows by Definitions 1.5.5 and 1.5.6. ■

Here is a well-known definition, which at once ensures the following corollary.

Definition 1.5.8 **L** *is a Tarski logic iff* **L** *is Reflexive, Transitive, and Monotonic.*

Corollary 1.5.3 *If* **L** *has static proofs, then* **L** *is a Tarski logic.*

It seems wise to mention uniformity, which takes its name from the Uniform Substitution rule. "Structurality" is another name for uniformity. Given the complications of Uniform Substitution rules at the predicative level—see [PP75]—let us first consider the propositional level. Let $s\colon \mathcal{S} \to \mathcal{W}_s$, a function that assigns a formula to every sentential letter. Extend s to all formulas, $s(A)$ being the result of replacing in A every sentential letter B by $s(B)$, and next to sets of formulas, $s(\Gamma) = \{s(A) \mid A \in \Gamma\}$. The Uniform Substitution rule states:

"if $\vdash A$, then $\vdash s(A)$".[35] I shall consider the generalized form and say that **L** is *propositionally uniform* iff $s(\Gamma) \vdash_{\mathbf{L}} s(A)$ holds whenever $\Gamma \vdash_{\mathbf{L}} A$ holds.[36]

The most obvious way to generalize Uniformity to the predicative level is by requiring that $\Gamma \vdash_{\mathbf{L}} A$ holds just in case it falls under a correct metalinguistic schema. However, the notion of a metalinguistic schema is not unambiguous. All depends on the metavariables one allows to occur in the schema. For example, if one allows for metavariables for any kind of entities that occur in $\mathcal{L}_s$, then, say, σ will be a metavariable for sentential letters. Suppose that $\vdash_{\mathbf{L}} p \supset p$. In order for **L** to be uniform on the so phrased criterion, it is sufficient that all formulas of the form $\sigma \supset \sigma$ are **L**-theorems, whereas propositional uniformity (or the Uniform Substitution rule) requires much more, viz. that all formulas of the form $A \supset A$ are **L**-theorems.

What is interesting about Propositional Uniformity is that it is an attempt to warrant the formal character of a logic. In order to generalize uniformity, for example to the predicative level, we need to refer to metalinguistic schemata. This reveals that uniformity is a relative matter, that it depends on the choice of the metalanguage. If one wants to generalize propositional uniformity to the predicative level in such a way that propositional uniformity is retained at the propositional level, one has to forbid that metavariables for sentential letters occur in the metalanguage. Actually, the simplest generalization is obtained by allowing only metavariables for individual constants and variables and for formulas, including mixed expressions such as $A(\alpha)$. The resulting uniformity is a very specific one, and by no means the most obvious one.

There is no need to make a specific choice at this point. Actually, it would be inappropriate to make a specific choice because this section concerns the general notion of static proofs. So uniformity should be considered as a *relative* property, depending on the chosen metalanguage. So let us define that a logic **L** is *uniform* with respect to a certain metalanguage just in case the following holds: $\Gamma \vdash_{\mathbf{L}} A$ iff it falls under a true statement $\Upsilon \vdash_{\mathbf{L}} \mathsf{A}$ of that metalanguage. That the schema $\Upsilon \vdash_{\mathbf{L}} \mathsf{A}$ is true obviously means that every statement $\Gamma' \vdash_{\mathbf{L}} A'$ which falls under the schema holds true.

To state the following theorem in a precise way, we need a further convention. Where a set of S-rules defines the static proofs of a logic and all metalinguistic formulas that occur in the S-rules belong to a certain metalanguage, I shall say that the static proofs are described in that metalanguage. If the static proofs are described in a certain metalanguage, then they are obviously also described in every richer metalanguage.

Theorem 1.5.7 *If* **L** *has static proofs that are described in a certain metalanguage, then* **L** *is uniform with respect to that metalanguage.*

Proof. Suppose that the antecedent is true and that $\Gamma \vdash_{\mathbf{L}} A$. It follows that there is a $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ. As the proof is obtained by applying S-rules, it is obvious that every formula in the proof can be replaced by a formula of the metalanguage in such a way that the resulting proof schema is correct.[37] ■

[35] This is different from the usual substitution rule, but has exactly the same effect: the rule in the text holds in a logic iff the usual rule holds.

[36] If the Deduction Theorem holds for **L**, then the generalized form holds for finite Γ whenever the usual form holds.

[37] For those not convinced that this is obvious, an algorithm for doing so is (implicitly) presented in Section 4.10.

Some take it that a logic has to be propositionally uniform in order to be formal, confusing handbook definitions with philosophical insight. Of course, everyone is free to define as he or she pleases. However, if by a formal logic we mean a logic **L** according to which $\Gamma \vdash_{\mathbf{L}} A$ iff A follows from Γ in view of the logical form of the members of Γ and of A, then some logics that are not propositionally uniform are nevertheless correctly called formal. I shall show this in Section 5.5.

I neither said nor implied that there is anything wrong with the Uniform Substitution rule. This rule holds for example in propositional **CL**, so it is perfectly all right to have Uniform Substitution as a primitive rule in a characterization of that logic. Moreover, it is an important property of **CL** and of many other logics that sentential letters have only those inferential properties which are common to all formulas of the language. I mention this explicitly because I have been misquoted too often in the past. But let us move on.

Lemma 1.5.2 *If* **L** *has static proofs, every line that occurs in a stage of a* $\mathcal{R}_{\mathbf{L}}$*-proof can be written as a finite string of a finite alphabet.*

Proof. This is obvious for the line numbers and formulas. The justification of a line contains a finite set of line numbers (in view of Corollary 1.5.1) and the name of a rule. So all line numbers involved can be written as a finite string of a finite alphabet and, as $\mathcal{R}_{\mathbf{L}}$ is a denumerable set, finite strings of a finite alphabet are sufficient to name all rules. The three elements of a line and the elements of the justification can obviously be separated by finitely many symbols.[38] ■

So we use a finite alphabet to write proof lines as finite strings. Actually, if the lemma would not hold, humans would not be able to write proofs.

There is a *positive test* for a logic **L** (**L** is *semi-decidable*, $Cn_{\mathbf{L}}(\Gamma)$ is semi-recursive) iff there is a mechanical procedure that, for every decidable Γ and A, leads after finitely many steps to the answer YES iff $\Gamma \vdash_{\mathbf{L}} A$ (but may not provide an answer at any finite point if $\Gamma \nvdash_{\mathbf{L}} A$).

Theorem 1.5.8 *If* **L** *has static proofs, then there is a positive test for* **L**.

Proof. Suppose that Γ is a decidable set of formulas and that $\Gamma \vdash_{\mathbf{L}} A$. In view of Definitions 1.5.4–1.5.6 and Theorem 1.5.2, there is a standard $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ and the last stage of this proof is a finite list of formulas. So this stage is a finite list of finite strings of a denumerable alphabet in view of Lemma 1.5.2.

All finite lists of finite strings of the alphabet in which proofs are written can be ordered into a list L. It is well-known (and easily seen) to be decidable whether a member of L is the last stage of a $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ. As some member of L is bound to be a $\mathcal{R}_{\mathbf{L}}$-proof of A from Γ, we shall find it after finitely many steps. ■

So it was proved that logics that have static proofs are Reflexive, Transitive, Monotonic, Uniform and Compact, and that there is a positive test for them. The reader may wonder whether all these Tarski-like properties are together sufficient to show that a logic has static proofs. And indeed, the following theorem is provable.

[38] Actually, no ambiguity arises if the separations are removed.

Theorem 1.5.9 *If* **L** *is Reflexive, Transitive, Monotonic, and Compact, and there is a positive test for it, then* there is a language *in which* **L** *has static proofs.*

The proof of the theorem is postponed to Section 2.7 because there I shall be able to illustrate it in terms of the logic **CLuN**, which is described in Section 2.2. The phrase "there is a language" is emphasized in the theorem because it introduces a major restriction as appears from the following theorem, which will also be proved in Section 2.7.

Theorem 1.5.10 *Some logics* **L***, defined over a language* $\mathcal{L}$ *are Reflexive, Transitive, Monotonic, Uniform and Compact, and there is a positive test for them, but do not have static proofs in* $\mathcal{L}$*.*

In other words, no recursive set of S-rules characterizes **L** if the proofs contain only formulas from $\mathcal{L}$. In view of the two theorems, I shall from now on prefer the phrase "static proofs" over the description in terms of the six properties. For one thing, it is shorter. Moreover, it is more precise in that it enables one to refer to a specific language.

Not all known logics have static proofs. A well-known exception is second order logic, which is not compact. An obvious example, taken from [BBJ02, p. 283] concerns the premise set comprising second order axioms for arithmetic (roughly Peano arithmetic plus the second order axiom of mathematical induction) together with all formulas of the form $\neg c = i$ (for $i \in \{0, 0', 0'', \ldots\}$ and c a constant that is added to the language of arithmetic). This set is inconsistent and hence $0 = 0'$ is derivable from it (by second order logic), but (on the supposition that arithmetic is consistent) $0 = 0'$ is not derivable from any subset of the premises.

That second order logic is not compact prevents it from having static proofs. Indeed, $0 = 0'$ cannot be derived in any stage from the aforementioned (infinite) premise set. Definition 1.5.1 does not rule out infinite lists of lines. Although one obviously cannot write down such proofs, we may consider them and reason about them, as we shall see in Section 4.7. However, the problem with second order logic lies elsewhere. Let $\mathcal{R}$ be a set of rules for second order logic. If L is a list of $\mathcal{R}$-correct lines, then $0 = 0'$ can only be the formula of a line of L if all premises are the formulas of *previous* lines of L (because every subset of the premise set is consistent). But if each premise is associated with a positive integer, there are no positive integers left to associate with $0 = 0'$. So $0 = 0'$ cannot be the formula of any line of L. Put differently, the definition of a static proof allows for infinite proofs but requires finitary S-rules. Second order logic requires infinitary rules.

Before leaving the matter, let me mention non-annotated proofs. Their stages are simply lists of formulas in which every formula is either a member of the premise set or derived from previous formulas in the list by one of the rules. The easy exercise of adjusting Definitions 1.5.1–1.5.6 to non-annotated proofs is left to the reader. I shall not pay much attention to non-annotated proofs in this book. The reason is that they are parasitic on annotated proofs. If the annotation is dropped, some stages become ambiguous in that a formula A may have been derived from different other (possibly empty) sets of formulas that precede A in the list. So in order to check whether a list of formulas is

$\mathcal{R}$-correct in view of a premise set, one has to show that it corresponds to a $\mathcal{R}$-correct stage of an annotated proof.

1.6 Semantics for Logics that Have Static Proofs

A semantics of a logic that has static proofs delineates a set of models. A model typically verifies certain formulas and falsifies others.

That a model M *verifies* A or, in other words, that A is *true* in M, will be abbreviated as $M \Vdash A$. If $M \nVdash A$, one says that A is *false* in M, and also that M *falsifies* A. It is handy to extend verification to sets: $M \Vdash \Gamma$ (M is a model of Γ) iff M verifies all $A \in \Gamma$. Obviously $M \nVdash \Gamma$ iff there is an $A \in \Gamma$ for which $M \nVdash A$.

It depends on the semantics what it precisely means that a model verifies a formula. So I shall specify this with every semantics I shall present. Many semantic systems in this book are two-valued. This means that, for every model M, there is a valuation function $v_M \colon \mathcal{W}_{\mathcal{O}} \to \{0, 1\}$. For all such semantics, $M \Vdash A$ iff $v_M(A) = 1$.

Given a semantics for a logic **L**, $\Gamma \vDash_{\mathbf{L}} A$ (A is a **L**-semantic consequence of Γ) iff all models of Γ verify A (that is: for all M, $M \nVdash \Gamma$ or $M \Vdash A$). $\vDash_{\mathbf{L}} A$ (A is **L**-valid, A is a valid formula of **L**) means that all **L**-models verify A. Where the context is unambiguous, I shall drop the reference to the logic.

In presenting a logic, it is wise to offer a syntactic characterization as well as a semantic characterization. The first refers to proofs, written by applying simple rules. Unfortunately, such rules do not always reveal the meaning of the logical symbols. Here is an example of the opaque character of rules (taken from [Bat87]). Let $\mathcal{L}$ be a propositional language with a single logical connective, the binary $\heartsuit$ (all sentential letters are well formed, and if A and B are well-formed, so is $(A \heartsuit B)$). The rules are defined by: one may derive B from $\{A_1, \ldots, A_n\}$ iff the following restriction is fulfilled for every sentential letter C: the number of occurrences of C in B is odd iff the number of occurrences of C in $A_1, \ldots, A_n$ is odd. Every child is able to construct proofs of this system and to check whether a list of lines is a correct proof stage. But what does $\heartsuit$ mean? Try a logic professor. Also try philosophers who claim that meaning is use. Here is the use, fellows. What is the meaning?

So this is where semantics has to play its role: it clarifies the meaning of logical symbols in a candid way. Unfortunately semantics is not very attractive from a computational point of view. Even the simplest reasoning about it refers to functions, infinite numbers of models, and, at the predicative level, domains that may be infinite or even uncountable.

Obviously one has to show that the syntactic characterization, often identified with the logic itself, corresponds exactly with the semantic characterization. This is done by proving that the logic is *sound* with respect to its semantics (viz. if $\Gamma \vdash A$ then $\Gamma \vDash A$) and that it is *complete* with respect to its semantics (viz. if $\Gamma \vDash A$ then $\Gamma \vdash A$).[39] If a logic is sound and complete with respect to a semantics, the semantics is said to be *characteristic* for the logic.

[39] Completeness with respect to its semantics is sometimes called a relative form of completeness. There are also absolute completeness forms, for example Lindenbaum-completeness and Post-completeness.

1.7 Classical Logic

CL will play an important part in this book. A first reason for this is that **CL** is taken to be the standard of deduction. Next, **CL** will be the only logic I shall use in the metalanguage—an "or" is a classical disjunction and a "not" is a classical negation. Finally, I shall also consider many logics that are close to **CL** and for which the central metatheoretic proofs, especially the Soundness and Completeness proofs, are similar to those for **CL**. Below I present an axiom system for **CL** as well as a (slightly unusual) semantics. Both concern the language $\mathcal{L}_s$, but their extension to $\mathcal{L}_S$ will also be considered. All this will be mainly useful for future reference.

CL is studied here in terms of closed formulas. Thus all axiom schemata and rules, and all semantic clauses contain closed formulas only. This settles at once the interpretation of ambiguous metavariables, for example that $\alpha \in \mathcal{V}_s$ and $\beta \in \mathcal{C}_s$ in axiom schema A∀.

I shall prove[40] several theorems, including that **CL** is complete with respect to the semantics. A first reason to do so is that the somewhat unusual formulation of the semantics in terms of a pseudo-language might make some readers suspicious. There is a second reason. Only a few obvious modifications are required to transform the completeness proof for **CL** into a completeness proof for most of the logics that will be considered in subsequent chapters.

CL is defined by the following axiom schemata and rules. In view of the logics presented in subsequent chapters, explicit definitions are avoided.

A⊃1	$A \supset (B \supset A)$
A⊃2	$(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$
A⊃3	$((A \supset B) \supset A) \supset A$
A∧1	$(A \wedge B) \supset A$
A∧2	$(A \wedge B) \supset B$
A∧3	$A \supset (B \supset (A \wedge B))$
A∨1	$A \supset (A \vee B)$
A∨2	$B \supset (A \vee B)$
A∨3	$(A \supset C) \supset ((B \supset C) \supset ((A \vee B) \supset C))$
A≡1	$(A \equiv B) \supset (A \supset B)$
A≡2	$(A \equiv B) \supset (B \supset A)$
A≡3	$(A \supset B) \supset ((B \supset A) \supset (A \equiv B))$
A¬1	$(A \supset \neg A) \supset \neg A$
A¬2	$A \supset (\neg A \supset B)$
A∀	$\forall \alpha A(\alpha) \supset A(\beta)$
A∃	$A(\beta) \supset \exists \alpha A(\alpha)$
A=1	$\alpha = \alpha$
A=2	$\alpha = \beta \supset (A(\alpha) \supset A(\beta))$
MP	From A and $A \supset B$ to derive B
R∀	To derive $\vdash A \supset \forall \alpha B(\alpha)$ from $\vdash A \supset B(\beta)$, provided β does not occur in either A or $B(\alpha)$.[41]

[40] Where only a proof outline is given, the proof is so standard that it can be found in or adjusted from any good logic handbook.

[41] The second turnstile in R∀ abbreviates the restriction that R∀ may only be applied if no application of Prem occurs in the path of $A \supset B(\beta)$. The first turnstile is written for reasons of tradition, but is actually useless. Similarly for the rule R∃.

R∃ To derive $\vdash \exists\alpha A(\alpha) \supset B$ from $\vdash A(\beta) \supset B$, provided β does not occur in either $A(\alpha)$ or B.

In view of Section 1.5, this axiom system defines $\Gamma \vdash_{\mathbf{CL}} A$ and $\vdash_{\mathbf{CL}} A$ for all $\Gamma \subseteq \mathcal{W}_s$ and $A \in \mathcal{W}_s$.

Theorem 1.7.1 CL *has static proofs.*

Proof. Immediate in view of the axiom system and Definition 1.5.6. ■

Corollary 1.7.1 CL *is reflexive, transitive, monotonic, compact, and there is a positive test for it.*

Actually, **CL** is not decidable, there is only a positive test for it—see any good handbook, for example [BBJ02].

Theorem 1.7.2 *If* $A_1, \ldots, A_n \vdash_{\mathbf{CL}} B$*, then* $A_1, \ldots, A_{n-1} \vdash_{\mathbf{CL}} A_n \supset B$. *(Deduction Theorem for* **CL***)*

Proof outline. Let $L = \langle C_1, \ldots, C_m \rangle$ be a proof of B $(= C_m)$ from $\{A_1, \ldots, A_n\}$. The list $L' = \langle A_n \supset C_1, \ldots, A_n \supset C_m \rangle$ can be transformed to a proof of $A_n \supset B$ from $\{A_1, \ldots, A_{n-1}\}$ by replacing every $A_n \supset C_i$ in L' by: (i) a proof of $A_n \supset A_n$ using A⊃1 and A⊃2 if C_i is A_n, (ii) a proof of $A_n \supset C_i$ from C_i using A⊃1 if C_i is an axiom or a member of $\{A_1, \ldots, A_{n-1}\}$, and (iii) a proof of $A_n \supset C_i$ from $A_n \supset (D \supset C_i)$ and $A_n \supset D$ using A⊃2 if C_i is obtained in L from $D \supset C_i$ and D by MP. ■

Betere veralgemening bewijzen?

Theorem 1.7.3 *If* $\Gamma \vdash_{\mathbf{CL}} B$ *and* $A \in \Gamma$*, then* $\Gamma - \{A\} \vdash_{\mathbf{CL}} A \supset B$. *(Generalized Deduction Theorem for* **CL***)*

Proof. Suppose that the antecedent is true. As **CL** is compact (Corollary 1.7.1), there are $C_1, \ldots, C_n$ such that $C_1, \ldots, C_n, A \vdash_{\mathbf{CL}} B$. By Theorem 1.7.2, $C_1, \ldots, C_n \vdash_{\mathbf{CL}} A \supset B$. As **CL** is monotonic (Corollary 1.7.1), $\Gamma - \{A\} \vdash_{\mathbf{CL}} A \supset B$. ■

We now turn to the semantics, which will be formulated in the language $\mathcal{L}_{\mathcal{O}}$ (which here is $\mathcal{L}_s$ extended with the set of pseudo-constants $\mathcal{O}$). A **CL**-model $M = \langle D, v \rangle$, in which D is a non-empty set and v an assignment function, is an interpretation of $\mathcal{W}_{\mathcal{O}}$ and hence of $\mathcal{W}_s$, which is what we are interested in. The following semantics is special in a second respect. In the usual **CL**-semantics the assignment assigns truth values (0 and 1) to sentential letters only, viz. $v\colon \mathcal{S} \to \{0, 1\}$. In the semantics below v assigns a truth value to all members of $\mathcal{W}_{\mathcal{O}}$—see C1. This modification does not have any effect for the **CL**-semantics—the clauses defining the valuation function, v_M, refer only to sentential letters, not to any other formulas. However, the modification will make life simpler when we come to semantic systems for other logics. The assignment function v is defined by:[42]

C1 $v\colon \mathcal{W}_{\mathcal{O}} \to \{0, 1\}$

[42]The restriction in C2 ensures that $\langle D, v \rangle$ is only a **CL**-model if $D = \{v(\alpha) \mid \alpha \in \mathcal{C} \cup \mathcal{O}\}$. In C3, $\wp(D^r)$ is the power set of the r-th Cartesian product of D.

C2 $v\colon \mathcal{C} \cup \mathcal{O} \to D$ (where $D = \{v(\alpha) \mid \alpha \in \mathcal{C} \cup \mathcal{O}\}$)
C3 $v\colon \mathcal{P}^r \to \wp(D^r)$

The valuation function $v_M\colon \mathcal{W}_{\mathcal{O}} \to \{0, 1\}$ determined by M is defined as follows:

C$\mathcal{S}$ where $A \in \mathcal{S}$, $v_M(A) = 1$ iff $v(A) = 1$
C$\mathcal{P}^r$ $v_M(\pi^r\alpha_1 \ldots \alpha_r) = 1$ iff $\langle v(\alpha_1), \ldots, v(\alpha_r)\rangle \in v(\pi^r)$
C= $v_M(\alpha = \beta) = 1$ iff $v(\alpha) = v(\beta)$
C¬ $v_M(\neg A) = 1$ iff $v_M(A) = 0$
C⊃ $v_M(A \supset B) = 1$ iff $v_M(A) = 0$ or $v_M(B) = 1$
C∧ $v_M(A \wedge B) = 1$ iff $v_M(A) = 1$ and $v_M(B) = 1$
C∨ $v_M(A \vee B) = 1$ iff $v_M(A) = 1$ or $v_M(B) = 1$
C≡ $v_M(A \equiv B) = 1$ iff $v_M(A) = v_M(B)$
C∀ $v_M(\forall\alpha A(\alpha)) = 1$ iff $\{v_M(A(\beta)) \mid \beta \in \mathcal{C} \cup \mathcal{O}\} = \{1\}$
C∃ $v_M(\exists\alpha A(\alpha)) = 1$ iff $1 \in \{v_M(A(\beta)) \mid \beta \in \mathcal{C} \cup \mathcal{O}\}$

$M \Vdash A$ (a **CL**-model M verifies A; A is true in a **CL**-model M) iff $v_M(A) = 1$. In view of Section 1.6, this semantics defines $\Gamma \vDash_{\mathbf{CL}} A$ and $\vDash_{\mathbf{CL}} A$ for all $\Gamma \subseteq \mathcal{W}_s$ and $A \in \mathcal{W}_s$.

As I shall introduce several non-standard logics, it is worth making the following somewhat elaborate comment. The identity of the object language, written as =, is defined by clause C=. All identities that occur in the semantics outside the expression $v_M(\alpha = \beta)$ are metalinguistic identities, for which I happen to use the same symbol. This cannot cause confusion in the present context—remember that the metalanguage is always classical—but it should be kept in mind for cases where a logic has a weaker identity than the classical one.

Up to this point, I have characterized **CL** syntactically as well as semantically with respect to the language $\mathcal{L}_s$. In view of later chapters, it is useful to extend this characterization to $\mathcal{L}_S$ in a specific way. For the axiomatic system, the simplest way is to proceed as follows. First replace in every axiom and rule every standard symbol by the corresponding classical symbol. Next add, for each standard symbol, an axiom that equates it to the classical symbol, viz. the eight axiom schemata $(A \supset B) \check{\equiv} (A \check{\supset} B)$, …, $\alpha = \beta \check{\equiv} \alpha \check{=} \beta$. For the semantics, we duplicate every clause defining the valuation value of a formula containing a logical symbol. Thus clause C= is matched by a clause C$\check{=}$ reading: $v_M(\alpha \check{=} \beta) = 1$ iff $v(\alpha) = v(\beta)$. In the rest of this section, I consider **CL** as defined for $\mathcal{L}_S$.

Theorem 1.7.4 *If* $\Gamma \vdash_{\mathbf{CL}} A$*, then* $\Gamma \vDash_{\mathbf{CL}} A$*.* *(Soundness of* **CL** *with respect to its semantics)*

Theorem 1.7.4 is proved by showing (i) that every axiom is a valid formula, (ii) that MP holds true in every model (if $M \Vdash A$ and $M \Vdash A \supset B$, then $M \Vdash B$), and (iii) that R∀ and R∃ hold for valid formulas (for example, for R∀, if $\vDash A \supset B(\beta)$ and β does not occur in either A or $B(\alpha)$, then $\vDash A \supset \forall\alpha B(\alpha)$). This is safely left to the reader.

In preparation of the proof of the strong completeness of **CL** with respect to its semantics, we need some definitions and lemmas. As I have to consider several languages and logics, the definitions are a trifle more complex than the usual ones.

Definition 1.7.1 Γ *is* **L**-deductively closed *in* $\mathcal{L}$ *iff* $Cn^{\mathcal{L}}_{\mathbf{L}}(\Gamma) \subseteq \Gamma$.

Definition 1.7.2 Γ *is* **L**-trivial *in* $\mathcal{L}$ *iff* $Cn^{\mathcal{L}}_{\mathbf{L}}(\Gamma) = \mathcal{W}$.

Where it is clear from the context which logic **L** and language $\mathcal{L}$ are intended, one says that Γ is trivial in case it is **L**-*trivial* in $\mathcal{L}$.

Definition 1.7.3 Γ *is* maximally **L**-non-trivial *in* $\mathcal{L}$ *iff* Γ *is* **L**-*non-trivial in* $\mathcal{L}$ *whereas, for every* $A \in \mathcal{W} - Cn^{\mathcal{L}}_{\mathbf{L}}(\Gamma)$, $\Gamma \cup \{A\}$ *is* **L**-*trivial in* $\mathcal{L}$.

Definition 1.7.4 *Where* Γ *is* **L**-*deductively closed in* $\mathcal{L}$, Γ *is* ω-complete *with respect to an existential quantifier* ξ *and a set of constants* $\mathcal{X}$ *iff* $A(\beta) \in \Gamma$ *for some* $\beta \in \mathcal{X}$ *whenever* $\xi\alpha A(\alpha) \in \Gamma$.

Let $\mathcal{L}_o$ be obtained from $\mathcal{L}_S$ by extending the set of constants $\mathcal{C}$ with the denumerable set of new constants $O = \{o_1, o_2, \ldots\}$—see Section 1.4. Let $\mathcal{W}^i_o$ $(i \geq 0)$ be those members of $\mathcal{W}_o$ in which occur no members of $\{o_{i+1}, o_{i+2}, \ldots\}$. Note that $\mathcal{W}_S \subset \mathcal{W}_o$ and that $\mathcal{W}_S \subset \mathcal{W}^i_o$ for all i.

Let L_0 be a list of all members of $\mathcal{W}_o$. We stepwise transform L_0 to a list L. After the ith step, the list L_0 will be transformed to a list L_i. Every step i is defined as follows: if $i > 1$ and the $(i-1)$th formula in L_{i-1} has the form $\check{\exists}\alpha A(\alpha)$, L_i is obtained by *inserting* $A(o_i)$ at the ith place; otherwise L_i is obtained by *moving* to the ith place the first member of $\mathcal{W}^{i-1}_o$ that occurs in L_{i-1} from the ith place on. Let $L = \langle B_1, B_2, \ldots\rangle$. Note that all members of $\mathcal{W}_o$ occur in L and that, in L, every formula of the form $\check{\exists}\alpha A(\alpha)$ is immediately followed by an instance $A(o_j)$ that does not occur in Γ or in a previous member of L.

Lemma 1.7.1 *Where* $\Gamma \subseteq \mathcal{W}_S$ *and* $A \in \mathcal{W}_S$, *if* $\Gamma \nvdash_{\mathbf{CL}} A$, *then there is a* $\Delta \subseteq \mathcal{W}_o$ *such that (i)* $\Gamma \subset \Delta$, *(ii)* $A \notin \Delta$, *(iii)* Δ *is* **CL**-*deductively closed in* $\mathcal{L}_o$, *(iv)* Δ *is maximally* **CL**-*non-trivial in* $\mathcal{L}_o$, *and (v)* Δ *is* ω-*complete with respect to* $\check{\exists}$ *and* $\mathcal{C} \cup O$.

Proof. Suppose that $\Gamma \nvdash_{\mathbf{CL}} A$. Define

$$\Delta_0 = Cn^{\mathcal{L}_o}_{\mathbf{CL}}(\Gamma)$$
$$\Delta_{i+1} = \begin{cases} Cn^{\mathcal{L}_o}_{\mathbf{CL}}(\Delta_i \cup \{B_{i+1}\}) & \text{if } A \notin Cn^{\mathcal{L}_o}_{\mathbf{CL}}(\Delta_i \cup \{B_{i+1}\}) \\ \Delta_i & \text{otherwise} \end{cases}$$
$$\Delta = \Delta_0 \cup \Delta_1 \cup \ldots .$$

I now prove the required properties of Δ. *Ad* (i): $\Gamma \subset \Delta$ in view of the reflexivity of **CL** (Corollary 1.7.1). *Ad* (ii): $A \notin \Delta$ in view of the supposition and the construction. *Ad* (iii): Suppose that $C \in \mathcal{W}_o$ and $\Delta \vdash_{\mathbf{CL}} C$. As **CL** is compact (Corollary 1.7.1), there is a finite $\Delta' \subseteq \Delta$ for which $\Delta' \vdash_{\mathbf{CL}} C$. It follows that there is an i for which $\Delta' \subseteq \Delta_i$,[43] whence $C \in \Delta_i$ in view of the construction. But then $C \in \Delta$ by the construction. So Δ is **CL**-deductively closed in $\mathcal{L}_o$. *Ad* (iv): As $A \notin \Delta$, $(A \supset C) \supset A \notin \Delta$ in view of A⊃3 and (iii). So, in view of (iii), $\Delta \nvdash_{\mathbf{CL}} (A \supset C) \supset A$ and, where B_i is $A \supset C$,

[43]There is a finite initial segment of L that contains all members of Δ'. As $\Delta' \subseteq \Delta$, there is no $C \in \Delta'$ for which $\Delta \cup \{C\} \vdash_{\mathbf{CL}} A$.

$\Delta_{i-1} \nvdash_{\mathbf{CL}} (A \supset C) \supset A$. But then $A \notin Cn^{\mathcal{L}_o}_{\mathbf{CL}}(\Delta_{i-1} \cup \{A \supset C\})$ in view of Theorem 1.7.3, and hence, by the construction,

$$A \supset C \in \Delta \,. \tag{1.1}$$

Note that (1.1) holds for all $C \in \mathcal{W}_o$. Consider a $D \in \mathcal{W}_o - \Delta$ and let D be B_j. As $D \notin \Delta$, $D \notin \Delta_j$ and hence, by the construction, $\Delta_{j-1} \cup \{D\} \vdash_{\mathbf{CL}} A$ and $\Delta \cup \{D\} \vdash_{\mathbf{CL}} A$. But then $\Delta \cup \{D\}$ is **CL**-trivial because (1.1) holds for all $C \in \mathcal{W}_o$. *Ad* (v): Suppose that $\check{\exists}\alpha C(\alpha) \in \Delta$ and that $C(\beta) \notin \Delta$ for all $\beta \in \mathcal{C} \cup O$. Let $\check{\exists}\alpha C(\alpha)$ be B_i, whence B_{i+1} is $C(o_{i+1})$. As $C(o_{i+1}) \notin \Delta$, $\Delta_i \cup \{C(o_{i+1})\} \vdash_{\mathbf{CL}} A$. By the compactness of **CL** (Corollary 1.7.1), there are $D_1, \ldots, D_n \in \Delta_i$ such that $C(o_{i+1}), D_1, \ldots, D_n \vdash_{\mathbf{CL}} A$. It follows that $\vdash_{\mathbf{CL}} C(o_{i+1}) \supset (D_1 \supset (\ldots \supset (D_n \supset A)\ldots))$. As o_{i+1} does not occur in A or in any of $D_1, \ldots, D_n$, $\vdash_{\mathbf{CL}} \check{\exists}\alpha C(\alpha) \supset (D_1 \supset (\ldots \supset (D_n \supset A)\ldots))$ in view of rule R$\exists$. But then $\check{\exists}\alpha C(\alpha), D_1, \ldots, D_n \vdash_{\mathbf{CL}} A$, and hence $A \in Cn^{\mathcal{L}_o}_{\mathbf{CL}}(\Delta_{i-1} \cup \{\check{\exists}\alpha C(\alpha)\})$, which contradicts $\check{\exists}\alpha C(\alpha) \in \Delta$. ■

Lemma 1.7.2 *If $\Delta \subseteq \mathcal{W}_o$ is* **CL***-deductively closed in $\mathcal{L}_o$, maximally* **CL***-non-trivial in $\mathcal{L}_o$, and ω-complete with respect to $\check{\exists}$ and $\mathcal{C} \cup O$, then there is a model M such that $v_M(A) = 1$ for all $A \in \Delta$ and $v_M(A) = 0$ for all $A \in \mathcal{W}_o - \Delta$.*

Proof outline. Suppose that the antecedent is true. For every $\alpha \in \mathcal{C} \cup O$, let $[\![\alpha]\!] = \{\beta \mid \alpha = \beta \in \Delta\}$.

Define a **CL**-model $M = \langle D, v\rangle$ as follows: $D = \{[\![\alpha]\!] \mid \alpha \in \mathcal{C} \cup O\}$ and

(i) for all $\alpha \in \mathcal{C} \cup O$: $v(\alpha) = [\![\alpha]\!]$,

(ii) $v(A) = 1$ iff $A \in \Delta \cap \mathcal{S}$,[44] and

(iii) where $\pi \in \mathcal{P}^r$, $v(\pi) = \{\langle [\![\alpha_1]\!], \ldots, [\![\alpha_r]\!]\rangle \mid \pi\alpha_1 \ldots \alpha_r \in \Delta\}$.

The proof that, for all $A \in \mathcal{W}_o$,

$$v_M(A) = 1 \text{ iff } A \in \Delta \tag{1.2}$$

proceeds by a straightforward induction on the complexity of A.[45] The basis of the induction is where $A \in \mathcal{S}$, or A has the form $\pi\alpha_1 \ldots \alpha_r$ with $\pi \in \mathcal{P}^r$ and $\alpha_1, \ldots, \alpha_r \in \mathcal{C} \cup O$, or A has the form $\alpha = \beta$ with $\alpha, \beta \in \mathcal{C} \cup O$. The proof that (1.2) holds for these A is obvious in view of the definition of M, and so are the cases of the induction step in view of the properties of Δ. ■

Theorem 1.7.5 *If $\Gamma \vDash_{\mathbf{CL}} A$, then $\Gamma \vdash_{\mathbf{CL}} A$.* *(Completeness of* **CL** *with respect to its semantics)*[46]

Proof. Immediate in view of Lemmas 1.7.1 and 1.7.2. ■

Answer to the puzzle from Section 1.6: ♡ is material equivalence.

[44] Remember that $v(A)$ does not have any effect on the valuation v_M if $A \in \mathcal{W}_{\mathcal{O}} - \mathcal{S}$.

[45] Unless otherwise specified, I shall take the complexity of a formula A to be identical to the number of logical symbols that occur in A with the exception of occurrences of identity. Thus the complexity of $\forall x((x = a \vee Px) \supset \neg Qx)$ is 4.

[46] As defined here, completeness is sometimes called “strong completeness” in order to distinguish it from weak completeness, which only concerns theoremhood (if $\vDash_{\mathbf{CL}} A$, then $\vdash_{\mathbf{CL}} A$). Weak completeness is rather irrelevant for this whole book.