

Chapter udf

Derivation Systems

This chapter collects general material on **derivation** systems. A textbook using a specific system can insert the introduction section plus the relevant survey section at the beginning of the chapter introducing that system.

prf.1 Introduction

fol:prf:int:
sec

Logics commonly have both a semantics and a **derivation** system. The semantics concerns concepts such as truth, satisfiability, validity, and entailment. The purpose of **derivation** systems is to provide a purely syntactic method of establishing entailment and validity. They are purely syntactic in the sense that a **derivation** in such a system is a finite syntactic object, usually a sequence (or other finite arrangement) of **sentences** or **formulas**. Good **derivation** systems have the property that any given sequence or arrangement of **sentences** or **formulas** can be verified mechanically to be “correct.”

The simplest (and historically first) **derivation** systems for first-order logic were *axiomatic*. A sequence of **formulas** counts as a **derivation** in such a system if each individual **formula** in it is either among a fixed set of “axioms” or follows from **formulas** coming before it in the sequence by one of a fixed number of “inference rules”—and it can be mechanically verified if a **formula** is an axiom and whether it follows correctly from other **formulas** by one of the inference rules. Axiomatic proof systems are easy to describe—and also easy to handle meta-theoretically—but **derivations** in them are hard to read and understand, and are also hard to produce.

Other **derivation** systems have been developed with the aim of making it easier to construct **derivations** or easier to understand **derivations** once they are complete. Examples are natural deduction, truth trees, also known as tableaux proofs, and the sequent calculus. Some **derivation** systems are designed especially with mechanization in mind, e.g., the resolution method is easy to implement in software (but its **derivations** are essentially impossible to

understand). Most of these other proof systems represent **derivations** as trees of **formulas** rather than sequences. This makes it easier to see which parts of a **derivation** depend on which other parts.

So for a given logic, such as first-order logic, the different **derivation** systems will give different explications of what it is for a **sentence** to be a *theorem* and what it means for a **sentence** to be **derivable** from some others. However that is done (via axiomatic **derivations**, natural deductions, sequent **derivations**, truth trees, resolution refutations), we want these relations to match the semantic notions of validity and entailment. Let's write $\vdash \varphi$ for “ φ is a theorem” and “ $\Gamma \vdash \varphi$ ” for “ φ is **derivable** from Γ .” However \vdash is defined, we want it to match up with \models , that is:

1. $\vdash \varphi$ if and only if $\models \varphi$
2. $\Gamma \vdash \varphi$ if and only if $\Gamma \models \varphi$

The “only if” direction of the above is called *soundness*. A **derivation** system is sound if **derivability** guarantees entailment (or validity). Every decent **derivation** system has to be sound; unsound **derivation** systems are not useful at all. After all, the entire purpose of a **derivation** is to provide a syntactic guarantee of validity or entailment. We'll prove soundness for the **derivation** systems we present.

The converse “if” direction is also important: it is called *completeness*. A complete **derivation** system is strong enough to show that φ is a theorem whenever φ is valid, and that $\Gamma \vdash \varphi$ whenever $\Gamma \models \varphi$. Completeness is harder to establish, and some logics have no complete **derivation** systems. First-order logic does. Kurt Gödel was the first one to prove completeness for a **derivation** system of first-order logic in his 1929 dissertation.

Another concept that is connected to **derivation** systems is that of *consistency*. A set of **sentences** is called inconsistent if anything whatsoever can be **derived** from it, and consistent otherwise. Inconsistency is the syntactic counterpart to unsatisfiability: like unsatisfiable sets, inconsistent sets of **sentences** do not make good theories, they are defective in a fundamental way. Consistent sets of **sentences** may not be true or useful, but at least they pass that minimal threshold of logical usefulness. For different **derivation** systems the specific definition of consistency of sets of **sentences** might differ, but like \vdash , we want consistency to coincide with its semantic counterpart, satisfiability. We want it to always be the case that Γ is consistent if and only if it is satisfiable. Here, the “if” direction amounts to completeness (consistency guarantees satisfiability), and the “only if” direction amounts to soundness (satisfiability guarantees consistency). In fact, for classical first-order logic, the two versions of soundness and completeness are equivalent.

prf.2 The Sequent Calculus

While many **derivation** systems operate with arrangements of **sentences**, the sequent calculus operates with *sequents*. A sequent is an expression of the form

$$\varphi_1, \dots, \varphi_m \Rightarrow \psi_1, \dots, \psi_n,$$

that is a pair of sequences of **sentences**, separated by the sequent symbol \Rightarrow . Either sequence may be empty. A **derivation** in the sequent calculus is a tree of sequents, where the topmost sequents are of a special form (they are called “initial sequents” or “axioms”) and every other sequent follows from the sequents immediately above it by one of the rules of inference. The rules of inference either manipulate the **sentences** in the sequents (adding, removing, or rearranging them on either the left or the right), or they introduce a complex **formula** in the conclusion of the rule. For instance, the \wedge L rule allows the inference from $\varphi, \Gamma \Rightarrow \Delta$ to $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$, and the \rightarrow R allows the inference from $\varphi, \Gamma \Rightarrow \Delta, \psi$ to $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$, for any Γ, Δ, φ , and ψ . (In particular, Γ and Δ may be empty.)

The \vdash relation based on the sequent calculus is defined as follows: $\Gamma \vdash \varphi$ iff there is some sequence Γ_0 such that every φ in Γ_0 is in Γ and there is a **derivation** with the sequent $\Gamma_0 \Rightarrow \varphi$ at its root. φ is a theorem in the sequent calculus if the sequent $\Rightarrow \varphi$ has a **derivation**. For instance, here is a **derivation** that shows that $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

$$\frac{\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L}{\Rightarrow (\varphi \wedge \psi) \rightarrow \varphi} \rightarrow R$$

A set Γ is inconsistent in the sequent calculus if there is a **derivation** of $\Gamma_0 \Rightarrow$ (where every $\varphi \in \Gamma_0$ is in Γ and the right side of the sequent is empty). Using the rule WR, any **sentence** can be **derived** from an inconsistent set.

The sequent calculus was invented in the 1930s by Gerhard Gentzen. Because of its systematic and symmetric design, it is a very useful formalism for developing a theory of **derivations**. It is relatively easy to find **derivations** in the sequent calculus, but these **derivations** are often hard to read and their connection to proofs are sometimes not easy to see. It has proved to be a very elegant approach to **derivation** systems, however, and many logics have sequent calculus systems.

prf.3 Natural Deduction

fol:prf.ntd:
sec

Natural deduction is a **derivation** system intended to mirror actual reasoning (especially the kind of regimented reasoning employed by mathematicians). Actual reasoning proceeds by a number of “natural” patterns. For instance, proof by cases allows us to establish a conclusion on the basis of a disjunctive premise, by establishing that the conclusion follows from either of the disjuncts. Indirect proof allows us to establish a conclusion by showing that its negation leads to a contradiction. Conditional proof establishes a conditional claim “if ... then ...” by showing that the consequent follows from the antecedent.

Natural deduction is a formalization of some of these natural inferences. Each of the logical connectives and quantifiers comes with two rules, an introduction and an elimination rule, and they each correspond to one such natural inference pattern. For instance, \rightarrow Intro corresponds to conditional proof, and \vee Elim to proof by cases. A particularly simple rule is \wedge Elim which allows the inference from $\varphi \wedge \psi$ to φ (or ψ).

One feature that distinguishes natural deduction from other **derivation** systems is its use of assumptions. A **derivation** in natural deduction is a tree of **formulas**. A single **formula** stands at the root of the tree of **formulas**, and the “leaves” of the tree are **formulas** from which the conclusion is derived. In natural deduction, some leaf **formulas** play a role inside the **derivation** but are “used up” by the time the **derivation** reaches the conclusion. This corresponds to the practice, in actual reasoning, of introducing hypotheses which only remain in effect for a short while. For instance, in a proof by cases, we assume the truth of each of the disjuncts; in conditional proof, we assume the truth of the antecedent; in indirect proof, we assume the truth of the negation of the conclusion. This way of introducing hypothetical assumptions and then doing away with them in the service of establishing an intermediate step is a hallmark of natural deduction. The formulas at the leaves of a natural **derivation** are called assumptions, and some of the rules of inference may “**discharge**” them. For instance, if we have a **derivation** of ψ from some assumptions which include φ , then the \rightarrow Intro rule allows us to infer $\varphi \rightarrow \psi$ and discharge any assumption of the form φ . (To keep track of which assumptions are discharged at which inferences, we label the inference and the assumptions it discharges with a number.) The assumptions that remain **undischarged** at the end of the **derivation** are together sufficient for the truth of the conclusion, and so a **derivation** establishes that its **undischarged** assumptions entail its conclusion.

The relation $\Gamma \vdash \varphi$ based on natural deduction holds iff there is a **derivation** in which φ is the last **sentence** in the tree, and every leaf which is **undischarged** is in Γ . φ is a theorem in natural deduction iff there is a **derivation** in which φ is the last **sentence** and all assumptions are **discharged**. For instance, here is a **derivation** that shows that $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

$$1 \frac{\frac{[\varphi \wedge \psi]^1}{\varphi} \wedge\text{Elim}}{(\varphi \wedge \psi) \rightarrow \varphi} \rightarrow\text{Intro}$$

The label 1 indicates that the assumption $\varphi \wedge \psi$ is **discharged** at the \rightarrow Intro inference.

A set Γ is inconsistent iff $\Gamma \vdash \perp$ in natural deduction. The rule \perp_I makes it so that from an inconsistent set, any **sentence** can be **derived**.

Natural deduction systems were developed by Gerhard Gentzen and Stanisław Jaśkowski in the 1930s, and later developed by Dag Prawitz and Frederic Fitch. Because its inferences mirror natural methods of proof, it is favored by philosophers. The versions developed by Fitch are often used in introductory

logic textbooks. In the philosophy of logic, the rules of natural deduction have sometimes been taken to give the meanings of the logical operators (“proof-theoretic semantics”).

prf.4 Tableaux

fol:prf:tab: While many derivation systems operate with arrangements of sentences, tableaux
sec operate with signed formulas. A signed formula is a pair consisting of a truth value sign (\mathbb{T} or \mathbb{F}) and a sentence

$$\mathbb{T}\varphi \text{ or } \mathbb{F}\varphi.$$

A tableau consists of signed formulas arranged in a downward-branching tree. It begins with a number of assumptions and continues with signed formulas which result from one of the signed formulas above it by applying one of the rules of inference. Each rule allows us to add one or more signed formulas to the end of a branch, or two signed formulas side by side—in this case a branch splits into two, with the two added signed formulas forming the ends of the two branches.

A rule applied to a complex signed formula results in the addition of signed formulas which are immediate sub-formulas. They come in pairs, one rule for each of the two signs. For instance, the $\wedge\mathbb{T}$ rule applies to $\mathbb{T}\varphi \wedge \psi$, and allows the addition of both the two signed formulas $\mathbb{T}\varphi$ and $\mathbb{T}\psi$ to the end of any branch containing $\mathbb{T}\varphi \wedge \psi$, and the rule $\varphi \wedge \psi\mathbb{F}$ allows a branch to be split by adding $\mathbb{F}\varphi$ and $\mathbb{F}\psi$ side-by-side. A tableau is closed if every one of its branches contains a matching pair of signed formulas $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$.

The \vdash relation based on tableaux is defined as follows: $\Gamma \vdash \varphi$ iff there is some finite set $\Gamma_0 = \{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ such that there is a closed tableau for the assumptions

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$$

For instance, here is a closed tableau that shows that $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	$\rightarrow\mathbb{F}1$
3.	$\mathbb{F}\varphi$	$\rightarrow\mathbb{F}1$
4.	$\mathbb{T}\varphi$	$\rightarrow\mathbb{T}2$
5.	$\mathbb{T}\psi$	$\rightarrow\mathbb{T}2$
	\otimes	

A set Γ is inconsistent in the tableau calculus if there is a closed tableau for assumptions

$$\{\mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$$

for some $\psi_i \in \Gamma$.

Tableaux were invented in the 1950s independently by Evert Beth and Jaakko Hintikka, and simplified and popularized by Raymond Smullyan. They

are very easy to use, since constructing a **tableau** is a very systematic procedure. Because of the systematic nature of **tableaux**, they also lend themselves to implementation by computer. However, a **tableau** is often hard to read and their connection to proofs are sometimes not easy to see. The approach is also quite general, and many different logics have **tableau** systems. **Tableaux** also help us to find **structures** that satisfy given (sets of) **sentences**: if the set is satisfiable, it won't have a closed **tableau**, i.e., any **tableau** will have an open branch. The satisfying **structure** can be “read off” an open branch, provided every rule it is possible to apply has been applied on that branch. There is also a very close connection to the sequent calculus: essentially, a closed **tableau** is a condensed **derivation** in the sequent calculus, written upside-down.

prf.5 Axiomatic Derivations

Axiomatic **derivations** are the oldest and simplest logical **derivation** systems. Its **derivations** are simply sequences of **sentences**. A sequence of **sentences** counts as a correct **derivation** if every **sentence** φ in it satisfies one of the following conditions:

fol:prf:axd:
sec

1. φ is an axiom, or
2. φ is an **element** of a given set Γ of **sentences**, or
3. φ is justified by a rule of inference.

To be an axiom, φ has to have the form of one of a number of fixed **sentence** schemas. There are many sets of axiom schemas that provide a satisfactory (sound and complete) **derivation** system for first-order logic. Some are organized according to the connectives they govern, e.g., the schemas

$$\varphi \rightarrow (\psi \rightarrow \varphi) \quad \psi \rightarrow (\psi \vee \chi) \quad (\psi \wedge \chi) \rightarrow \psi$$

are common axioms that govern \rightarrow , \vee and \wedge . Some axiom systems aim at a minimal number of axioms. Depending on the connectives that are taken as primitives, it is even possible to find axiom systems that consist of a single axiom.

A rule of inference is a conditional statement that gives a sufficient condition for a **sentence** in a **derivation** to be justified. Modus ponens is one very common such rule: it says that if φ and $\varphi \rightarrow \psi$ are already justified, then ψ is justified. This means that a line in a **derivation** containing the **sentence** ψ is justified, provided that both φ and $\varphi \rightarrow \psi$ (for some **sentence** φ) appear in the **derivation** before ψ .

The \vdash relation based on axiomatic **derivations** is defined as follows: $\Gamma \vdash \varphi$ iff there is a **derivation** with the **sentence** φ as its last formula (and Γ is taken as the set of **sentences** in that derivation which are justified by (2) above). φ is a theorem if φ has a **derivation** where Γ is empty, i.e., every **sentence** in the derivation is justified either by (1) or (3). For instance, here is a **derivation** that shows that $\vdash \varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi))$:

1. $\psi \rightarrow (\psi \vee \varphi)$
2. $(\psi \rightarrow (\psi \vee \varphi)) \rightarrow (\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi)))$
3. $\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi))$

The **sentence** on line 1 is of the form of the axiom $\varphi \rightarrow (\varphi \vee \psi)$ (with the roles of φ and ψ reversed). The sentence on line 2 is of the form of the axiom $\varphi \rightarrow (\psi \rightarrow \varphi)$. Thus, both lines are justified. Line 3 is justified by modus ponens: if we abbreviate it as θ , then line 2 has the form $\chi \rightarrow \theta$, where χ is $\psi \rightarrow (\psi \vee \varphi)$, i.e., line 1.

A set Γ is inconsistent if $\Gamma \vdash \perp$. A complete axiom system will also prove that $\perp \rightarrow \varphi$ for any φ , and so if Γ is inconsistent, then $\Gamma \vdash \varphi$ for any φ .

Systems of axiomatic **derivations** for logic were first given by Gottlob Frege in his 1879 *Begriffsschrift*, which for this reason is often considered the first work of modern logic. They were perfected in Alfred North Whitehead and Bertrand Russell's *Principia Mathematica* and by David Hilbert and his students in the 1920s. They are thus often called “Frege systems” or “Hilbert systems.” They are very versatile in that it is often easy to find an axiomatic system for a logic. Because **derivations** have a very simple structure and only one or two inference rules, it is also relatively easy to prove things *about* them. However, they are very hard to use in practice, i.e., it is difficult to find and write proofs.

Photo Credits

Bibliography