

Finding Derivations

Chapters 1 and 2 gave every connective a rule for using it and a rule for proving it, and a derivation, once written, can be checked line by line with no idea at all. Finding one is another matter. An assumption may be any statement whatever, and nothing in the rules says which one to make; yet every Idea so far found its derivation with a handful of moves. *“You know every propositional rule. Given a sequent, how do you find a derivation, or decide to stop looking?”* The chapter answers in three steps. Section 3.1 puts the moves in order as one method, says why the order loses nothing, and watches the method succeed and fail. Section 3.2 shows that the **task** shapes the search, and names the syntactic idea behind every task: an **inconsistent** list, one from which $\perp$ is derivable. Section 3.3 starts from a paragraph of mathematics instead of a sequent. The chapter ends on a question its methods cannot answer: a search that fails proves nothing, so how could anyone show that something is **not** derivable?

What you need. Chapters 1 and 2: the rules for every connective; derivations with boxes and the scope rule (Definition 1.4.2); derived rules (Definition 2.3.1) and Lemma 2.3.3; the classical rules of Section 2.5; and from Section 1.1 the counterexample set (Definition 1.1.3) and the move *try to break it*. Steps about “every” and “some”, such as “say $m = 2k$ ”, are used as ordinary reasoning, as in Chapter 2, until Chapter 5 gives them their rules.

Time. About 15 hours on the core track (all of A, one B and one C per section, the exit checks); about 25 hours for everything.

Minimum before *Sets, Relations, and Cardinality*. Chapters 1–6 and 8, core; Sections 7.1–7.2 before its Chapter 3.

3.1.1 Roadmap

- **How to Find a Derivation.** The moves of Chapters 1 and 2 put in order as one method, the reason the order loses nothing, four refinements including the detour rule, searches that succeed and one that fails, and every rule on one card.
- **Four Kinds of Task.** Theorems, equivalences, inconsistent lists and claims whose verdict is withheld: how each task starts its search, and why all four are one question, whether a list derives $\perp$.
- **From Argument to Derivation.** Extracting the argument from a paragraph, naming the facts it leaves unsaid, writing a proof and its derivation side by side, and a library of standard sequents grouped by what they are for.

3.1.2 Named moves

- **Work from both ends.** The method: read the skeleton off the goal, use the resources by their main connectives, keep the classical step for last. (Section 3.1)
- **Find the goal among the resources.** If the current goal is already available, cite it or copy it by R. (Section 3.1)
- **Derive what needs nothing new before opening a box.** Lines written outside the boxes are available in all of them. (Section 3.1)
- **Look first for a clash partner already on a line.** For the goal $\perp$, aim at the statement whose negation you already have. (Section 3.1)
- **No detours.** Never build a line by an introduction rule and take it apart at once by the matching elimination. (Section 3.1)
- **Let the task choose the search.** A theorem starts at the goal, an equivalence needs

two derivations, an inconsistency aims at a clash. (Section 3.2)

- **Try to break it first.** Before searching, try to make the counterexample set true. An attempt that is forced into a clash at every turn is usually a derivation read backwards: its forced statements suggest the boxes, and its clash the $\perp$. (Section 3.2)
- **Name the unstated fact.** An argument in prose leaves premises unsaid; write each one down and check it. (Section 3.3)
- **Two columns.** Write the proof beside the derivation it mirrors, and see which lines the prose leaves silent. (Section 3.3)

3.-1.3 Summary

The search method reads boxes off the goal and eliminations off the resources, and its goal steps and its classical step never turn a gap that can be filled into one that cannot (the fold of Section 3.1). Postponing a disjunction and the four refinements make it efficient. Every task in Section 3.2 is a question about derivability, and every question about derivability is a question about whether a list is **inconsistent** (Proposition 3.2.2): deriving φ from Γ is showing the counterexample set $\Gamma, \neg\varphi$ inconsistent. An inconsistent list derives every statement (Proposition 3.2.1). Arguments in prose become sequents once their unstated facts are named, and the library of Section 3.3 sorts the standard sequents by what they are for.

Every method of the chapter can show that something **is** derivable, and none can show that something is **not**: a claim that breaks, or a search that stalls, has not been shown underivable. That is Chapter 4's question.

3.-1.4 Terms

- **consistent list**: a list from which $\perp$ is not derivable (Definition 3.2.1).
- **counterexample set** (recalled): the premises together with the negated conclusion (Definition 1.1.3).
- **current goal**: the written line below a gap (Definition 3.1.1).
- **gap**: a stretch of a partial derivation still missing (Definition 3.1.1).
- **inconsistent list**: a list from which $\perp$ is derivable (Definition 3.2.1).
- **partial derivation**: a derivation with gaps still to fill (Definition 3.1.1).
- **provably equivalent** (recalled): each derivable from the other (Definition 2.4.2).
- **resources**: the lines available at a gap's current goal (Definition 3.1.1).
- **theorem (of logic)**: a statement derivable from no premises, $\vdash \varphi$ (Definition 1.2.3).