

4.6 Hunting for a Refuting Row

Section 4.5 reduced every question about rows to one: is this list unsatisfiable? A truth table answers it by brute force, and with n atoms it has 2^n rows. Working backwards from F (Example (Working backwards from F)) is faster: assume the conclusion false and follow what is forced. But where a value leaves a choice, as a false conjunction or a true disjunction does, it only says “split into cases”, and keeps no record of the cases.

This section turns working backwards into a method that keeps that record: take the forced steps first, and when nothing is forced, split into cases and follow every case. On paper the method is a tree. A branch that closes is a case with no row; a branch that stays open hands us a row. The answers are about truth, “valid” or “here is a refuting row”; what they say about derivations is the next section’s business.

4.6.1 The rules of the search

To decide $\Gamma \models \psi$, we look for one row that makes every formula of the counterexample set $\Gamma, \neg\psi$ true; the entailment holds exactly when there is none. To decide whether a list Δ is satisfiable, we look for one row that makes all of Δ true. Requirements on the row are kept as formulas: “ φ must be true” is written φ , and “ φ must be false” is written $\neg\varphi$.

Quick check. From the tables of (V3)–(V5), counting each way as a condition on one or both parts: in how many ways can $\varphi \wedge \psi$ be true? In how many ways can it be false? In how many ways can $\varphi \rightarrow \psi$ be false? In how many ways can it be true?

Solution. $\varphi \wedge \psi$ is true in one way (both parts true) and false in two (φ false, or ψ false; the row with both false is covered by either). $\varphi \rightarrow \psi$ is false in one way (φ true and ψ false) and true in two (φ false, or ψ true).

Each requirement on a compound formula is traded for requirements on its parts, read off its clause. These are the **search rules**; first the requirements met in only one way.

On the branch	Its requirement, read off	Adds
$\varphi \wedge \psi$	(V3): both parts true	φ, ψ
$\neg(\varphi \vee \psi)$	(V4): both parts false	$\neg\varphi, \neg\psi$
$\neg(\varphi \rightarrow \psi)$	(V5): hypothesis true, conclusion false	$\varphi, \neg\psi$
$\neg\neg\varphi$	(V2) twice: φ true	φ

Then the requirements met in two ways; each splits the search into two cases.

On the branch	Its requirement, read off	Adds
$\varphi \vee \psi$	(V4): some part true	φ or ψ
$\neg(\varphi \wedge \psi)$	(V3): some part false	$\neg\varphi$ or $\neg\psi$

On the branch	Its requirement, read off	Adds
$\varphi \rightarrow \psi$	(V5): hypothesis false, or conclusion true	$\neg\varphi$ or ψ
$\varphi \leftrightarrow \psi$	(V6): the parts agree	φ, ψ or $\neg\varphi, \neg\psi$
$\neg(\varphi \leftrightarrow \psi)$	(V6): the parts differ	$\varphi, \neg\psi$ or $\neg\varphi, \psi$

An atom, a negated atom, and $\neg\perp$ (true in every row) need no rule. A list of requirements that holds some formula together with its negation, or holds $\perp$, describes no row at all: it is **closed**. The one-way rules are the forced steps of working backwards from F, with $\neg\neg$ added; the two-way rules are where it split into cases, with $\leftrightarrow$ added.

The two ways of a split need not exclude each other (a row may make both φ and ψ true); they need only **cover** every row that makes the formula true.

4.6.2 The search, drawn as a tree

Definition 4.6.1. Search Tree

Let Δ be a finite list of formulas. A **search tree** for Δ is built in stages. It begins as one node, the **root**, carrying the formulas of Δ . At each stage, choose an open branch B and a formula χ on B that is not an atom, a negated atom, $\perp$ or $\neg\perp$, and that is not yet **used** on B ; then, below the last node of B , add χ 's rule: one new node with the formulas of its one way, or two new nodes side by side with the formulas of its two ways. On each new branch, χ (and every formula equal to it) is now used: a formula is **used** on a branch exactly when the branch passes through a node added by its rule, so it stays used on every branch extending that one. Being used is a property of a formula **on a branch**: a formula at the root may be used on one branch and not on another.

A **branch** is a path from the root down to a node with nothing below it, and its formulas are those on its nodes. A branch is **closed** if it holds some formula together with its negation, or holds $\perp$; otherwise it is **open**. An open branch is **finished** if every formula on it, other than an atom, a negated atom or $\neg\perp$, is used on it. A search tree is **finished** if each of its branches is closed or finished.

In words: a search tree is a record of a backwards computation with every choice followed. A closed branch is a case with no row; a finished open branch is a case that has survived every requirement.

We draw search trees with three marks. A closed branch ends in $\times$, and a finished open branch in $\circ$. A formula is ticked ($\checkmark$) once its rule has been applied below it on some branch; the nodes its rule added show on which branches it is used. The formulas that a one-way rule adds are written straight under the stack they extend, with no new edge.

Example. A tree that closes

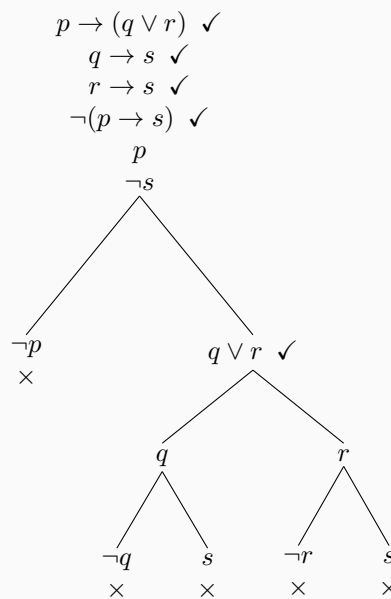
Decide whether $p \rightarrow (q \vee r)$, $q \rightarrow s$, $r \rightarrow s \models p \rightarrow s$.

Solution. This is the list of Section 3.2's exit check (Exercise (C1: Exit check: the counterexample set, used later)), which derived the claim; now a search shows it valid.

The root carries the counterexample set: the three premises and $\neg(p \rightarrow s)$.

- Stage 1, the one-way rule for $\neg(p \rightarrow s)$: add p , $\neg s$.
- Stage 2, $p \rightarrow (q \vee r)$: $\neg p$ or $q \vee r$. The branch $\neg p$ closes against p .
- Stage 3, $q \vee r$: q or r .
- Stage 4, on the branch q : $q \rightarrow s$ gives $\neg q$, which closes against q , or s , which closes against $\neg s$.
- Stage 5, on the branch r : $r \rightarrow s$ gives $\neg r$ or s , both closed.

(The branch r closes without $q \rightarrow s$, so that formula is never used there.) Every branch is closed: no row makes the counterexample set true, and the entailment holds.



A tick marks a formula whose rule was applied below it: $q \rightarrow s$ was used on the branch through q , and $r \rightarrow s$ on the branch through r . Each branch closed before the other formula was needed.

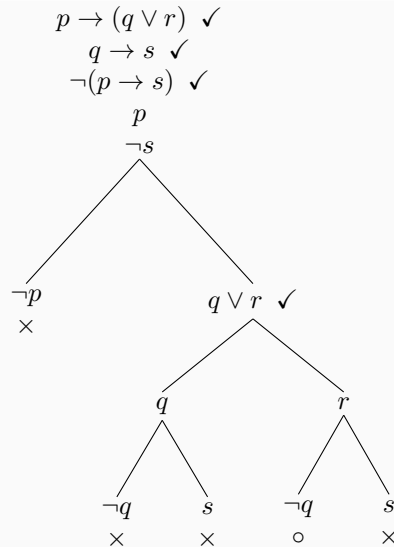
Five closed branches replace the $2^4 = 16$ rows of a table. Read as a proof by cases, the tree says: p is forced, so $q \vee r$; in the case q , s follows, against $\neg s$; in the case r , likewise.

Example. A branch that stays open

Decide whether $p \rightarrow (q \vee r)$, $q \rightarrow s \models p \rightarrow s$.

Solution. The first three stages are the same as before. On the branch q , the rule of $q \rightarrow s$ closes both sides, as before.

On the branch r , at a stage of its own, $q \rightarrow s$ gives $\neg q$ or s . (A formula used on one branch may be used again on another.) The side s closes against $\neg s$; the side $\neg q$ does not. That branch carries $p \rightarrow (q \vee r)$ (used), $q \rightarrow s$ (used), $\neg(p \rightarrow s)$ (used), p , $\neg s$, $q \vee r$ (used), r and $\neg q$: it is open and finished.



Its row: the atoms on it, p and r , are true; the atoms that stand negated on it, q and s , are false. Check against the root. $p \rightarrow (q \vee r)$ is $T \rightarrow T$, true. $q \rightarrow s$ has a false hypothesis, so it is true. $p \rightarrow s$ is $T \rightarrow F$, false. So this row refutes the claim.

The premise $r \rightarrow s$ of the first example is exactly what closed this branch. Dropping it breaks the claim, as replacing it by $r \rightarrow t$ did in part (d) of Exercise (C1: Exit check: the counterexample set, used later), and now the search finds the row. The search may stop at the first finished open branch: one refuting row settles the question.

Quick check. On the open branch of the second tree, check that its row makes every formula on the branch true, the used ones included.

Solution. p , r , $\neg q$ and $\neg s$ are true; $q \vee r$ is true, by r ; $p \rightarrow (q \vee r)$ is true, its conclusion being true; $q \rightarrow s$ is true, its hypothesis being false; and $\neg(p \rightarrow s)$ is true, since $p \rightarrow s$ is false.

4.6.3 What a closed tree proves

Each rule passes a row on. If a row makes every formula on a branch true and a rule is applied there, the row makes true all the formulas of at least one new node, since each rule was read off its table. So a row making the root true runs down some branch to its end with every formula true there. A closed branch holds φ and $\neg\varphi$, or $\perp$, which no row makes true.

An open finished branch hands over a row. Make the atoms that stand on it true and all other atoms false. A negated atom on it is then true, since its atom is not on the open branch; every other formula on it is used, so the branch carries the shorter formulas of one of its ways, and it is true by its clause once they are. From the shortest up, every formula on the branch is true.

Proposition 4.6.1. The Search Decides

Let Δ be a finite list of formulas.

- (a) Every search tree for Δ can be continued until it is finished, and it has only finitely many stages.
- (b) If every branch of a search tree for Δ is closed, then Δ is unsatisfiable.
- (c) If a search tree for Δ has an open finished branch B , then the valuation v_B that makes the atoms on B true and every other atom false makes every formula on B true, in particular every formula of Δ .

So, for a list Γ and a formula ψ : a search tree for $\Gamma, \neg\psi$, continued as in (a), either closes, and then $\Gamma \models \psi$, or has an open finished branch, whose row refutes $\Gamma \models \psi$.

In words: the search always ends; a closed tree proves unsatisfiability, and an open finished branch exhibits a satisfying row. Since (b) and (c) cannot both apply to finished trees of one list, every order of moves gives the same verdict.

Idea.

- (a) Every new formula is shorter than the one whose rule added it, and is a subformula of a root formula or the negation of one. (b) and (c) are the two observations, made exact by induction (on the stages for (b), on the length of a formula for (c)). The proof is in the fold below.

🕒 **Second reading** · *The proof: why the search decides*

Proof.

- (a) Let N be the number of formulas that are subformulas of formulas of Δ , or negations of such. Every formula ever added to the tree is one of these. Each rule adds immediate parts of the used formula, or negations of them, or, for $\neg\neg\varphi$, the formula φ ; and if the used formula is a subformula of a root formula or the negation of one, then so are the formulas it adds. Once a formula is used on a branch, it and every formula equal to it stay used on every branch below, so a stage below a given node never uses a formula already used above it. Hence each of the N formulas is used at most once along a branch, and a branch has at most N stages below the root. A stage adds at most two nodes, below one branch, so a branch splits at most N times, and the tree never has more than 2^N branches, each with at most $N + 1$ nodes. Each stage adds a node, so there are only finitely many stages. When no stage is possible, no open branch carries an unused formula other than an atom, a negated atom, $\perp$ or $\neg\perp$; since $\perp$ is on no open branch, every branch is closed or finished.
- (b) *Claim:* at every stage, if a valuation v makes every formula of Δ true, then some branch has all its formulas true under v . At the start this is the root. Suppose it holds before a stage, and the stage applies χ 's rule below the chosen branch B' . A branch B all of whose formulas are true under v is either another branch, which the stage leaves unchanged, or B' itself. In the second case v makes χ true, since χ is on B' , and by the clause of Definition 4.4.2 from which the rule was read, v makes true every formula of one new node. One rule at a time:
 - $\varphi \wedge \psi$ true: φ and ψ true, by (V3).
 - $\neg(\varphi \vee \psi)$ true: $\varphi \vee \psi$ false, so $\neg\varphi$ and $\neg\psi$ true, by (V2) and (V4).
 - $\neg(\varphi \rightarrow \psi)$ true: $\varphi \rightarrow \psi$ false, so φ and $\neg\psi$ true, by (V2) and (V5).

- $\neg\neg\varphi$ true: $\neg\varphi$ false, so φ true, by (V2) twice.
- $\varphi \vee \psi$ true: φ true or ψ true, by (V4).
- $\neg(\varphi \wedge \psi)$ true: $\varphi \wedge \psi$ false, so $\neg\varphi$ true or $\neg\psi$ true, by (V2) and (V3).
- $\varphi \rightarrow \psi$ true: φ false or ψ true, so $\neg\varphi$ true or ψ true, by (V5) and (V2).
- $\varphi \leftrightarrow \psi$ true: the parts agree, so both are true, or $\neg\varphi$ and $\neg\psi$ are both true, by (V6) and (V2).
- $\neg(\varphi \leftrightarrow \psi)$ true: the parts differ, so φ and $\neg\psi$ are true, or $\neg\varphi$ and ψ are, by (V2) and (V6).

So the new branch through that node has all its formulas true under v , and the claim holds after the stage. By Axiom 4.2.1 on the number of stages, it holds for the tree at hand. A closed branch holds φ and $\neg\varphi$, or $\perp$, and v cannot make all of these true, by (V2) and (V1). So if every branch is closed, no valuation makes every formula of Δ true: Δ is unsatisfiable.

(c) Let B be open and finished. We prove that $v_B(\chi) = T$ for every formula χ on B , by Theorem 4.2.2 on the length of χ : assume it for every formula on B shorter than χ .

- An atom p on B is true, by the definition of v_B .
- $\perp$ is not on B , since B is open.
- $\neg\perp$ is true, by (V1) and (V2).
- A negated atom $\neg p$: p is not on B , else B would be closed, so $v_B(p) = F$ and $\neg p$ is true.
- Any other χ is used on B , since B is finished. So B passes through a node carrying the formulas of one of χ 's ways. Each of them is shorter than χ : by the count of symbols in Section 4.3, $\neg\varphi$ has one symbol more than φ , and $(\varphi \circ \psi)$ has the symbols of φ and ψ plus 3, so each of φ , ψ , $\neg\varphi$, $\neg\psi$ is shorter than $(\varphi \circ \psi)$ and than its negation, and φ is shorter than $\neg\neg\varphi$. So each is true under v_B by the hypothesis, and then χ is true by the clause of its row in the bullets of (b), read from right to left.

The formulas of Δ are at the root, which is on B , so all of them are true under v_B .

The last sentence of the statement: apply (a), and then (b) or (c), to the list $\Delta = \Gamma, \neg\psi$. By Section 4.5, $\Gamma \models \psi$ exactly when this list is unsatisfiable, and a valuation making it true makes every formula of Γ true and ψ false.

■

Warning. A split needs **both** sides. Had we said, in the first tree, “suppose q is true”, closed that branch and stopped, we would have shown only that no refuting row makes q true; the rows with q false and r true were never examined. A clash reached under a guess refutes the guess, not the claim. The converse slip: an open branch proves something only when it is **finished**. While a formula on it is unused, it may still close.

4.6.4 Choosing the next move

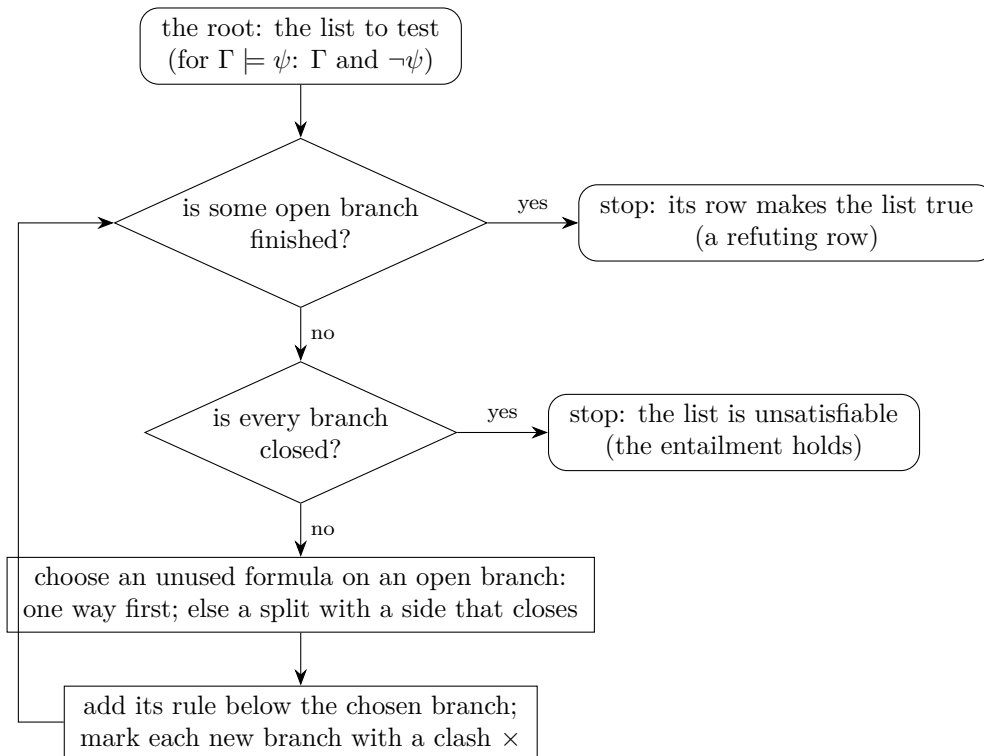
Every order gives the same verdict; the work differs. Five tactics keep a tree small.

1. **The conclusion first.** $\neg\psi$ is often one way, and it fixes the values that make the premises bite.
2. **One way before two ways.** A one-way rule adds to every branch at once; a split doubles

the work below it.

3. **Close early.** Before a split, look for a formula whose negation is already on the branch; after it, close every branch you can at once.
4. **Split where a side closes at once.** Among formulas that must split, prefer one with a side that clashes immediately.
5. **Stop at the first finished open branch.** One refuting row is enough.

The method as a chart:



The order of the tests decides nothing about the verdict ([Proposition 4.6.1](#)), only about the work.

Quick check. Search [Example \(A tree that closes\)](#) again, applying the rule of $p \rightarrow (q \vee r)$ before that of $\neg(p \rightarrow s)$. What changes, and what does not?

Solution. The one-way node p , $\neg s$ must now be written under each open side of the first split, at a stage of its own when that side is chosen, so the tree is larger. The side $\neg p$ closes only after p has been added under it. Every branch still closes, and by [Proposition 4.6.1](#) the verdict cannot change.

4.6.5 Testing a list for satisfiability

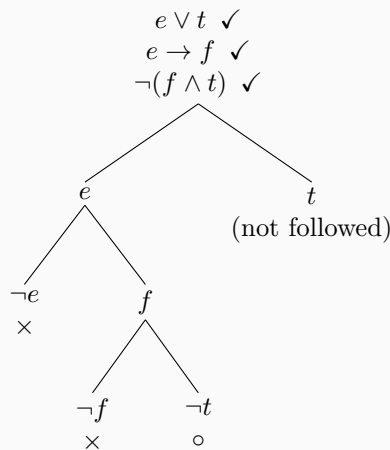
The same tree works for any finite list. For a list of claims about an integer, an open finished branch gives a row, and often a number.

Example. Three claims about an integer

Let n be an integer. Can these three claims all be true?

- (1) n is even or n is a multiple of 3.
- (2) If n is even, then n is a multiple of 4.
- (3) n is not both a multiple of 4 and a multiple of 3.

Solution. Take the atoms e : “ n is even”, t : “ n is a multiple of 3” and f : “ n is a multiple of 4”. The list is $e \vee t$, $e \rightarrow f$, $\neg(f \wedge t)$. No rule is one way, so we split $e \vee t$: e or t . Under e , the rule of $e \rightarrow f$ gives $\neg e$, which closes, or f . Under f , the rule of $\neg(f \wedge t)$ gives $\neg f$, which closes, or $\neg t$. That branch, $e, f, \neg t$ with every compound formula used, is open and finished.



A number with its row (e , f true, t false) is $n = 4$: even, a multiple of 4, not a multiple of 3. All three claims hold for it. The branch t need not be followed.

The search treats e , f and t as independent letters. Here both rows that satisfy the list, $e, f, \neg t$ and $\neg e, \neg f, t$ (the branch t , followed to the end), happen to be rows of integers, 4 and 3; nothing in the search guarantees it. A row such as $e = F$, $f = T$, which no integer has (a multiple of 4 is even), is a row like any other to the search, as the warning at the end of Section 4.4 says.

Section 3.2 asked whether a list is **inconsistent**; the search asks whether it is **unsatisfiable**, and Sections 4.7 and 4.8 prove that the two questions always have the same answer.

Quick check. Section 3.2 derived $\perp$ from the list $p \rightarrow q$, $q \rightarrow \neg p$, p (Exercise (B4: Derive $\perp$, or make every member true, used later) (a)). Is it satisfiable?

Solution. No. The rule of $p \rightarrow q$ gives $\neg p$, which closes against p , or q . Under q , the rule of $q \rightarrow \neg p$ gives $\neg q$, which closes, or $\neg p$, which closes against p . Every branch is closed, so by [Proposition 4.6.1](#) (b) the list is unsatisfiable.

🕒 **Second reading** · *A tree is a table with rows merged*

Each branch of a search tree stands for the rows that agree with its atoms and negated atoms and make true every formula split above it: a branch carrying p and $\neg s$, with no split above it, stands for every row with p true and s false. A closed branch stands for no row at all. A split keeps the rows of its branch that make the split formula true and divides them into two groups, which may overlap; a row of the branch that makes the formula false lies on neither side and is struck out at the split itself. So a tree whose branches all close is a truth table whose rows have been grouped and struck out a group at a time. In the first tree of this section, the root's p and $\neg s$ leave four of the sixteen rows on p, q, r, s as candidates; the branch through q and its closed leaves strike out the two of those with q true, the branch through r the one left with q false and r true, and the row with q and r both false lies on neither side of the split of $p \rightarrow (q \vee r)$, which it makes false. Between them, the root's atoms, the splits and the five closed branches strike out all sixteen rows as candidates for a refuting row, and nobody had to write any of them down.

4.6.6 The withheld verdicts

Section 3.2's derive-or-break items withheld the verdict, and a claim that resisted both deriving and breaking by hand stayed open. The search settles each **semantically**: a closed tree shows the claim valid, an open finished branch gives a refuting row, always in finitely many steps.

Here are the four claims that Exercise B3 of Section 3.2 broke (Exercise (B3: Derive or break, used later), used later), read as entailments, each with the row its search returns. $\times$ marks a side that closes.

Claim	Refuting row	The search
(a) $\neg p \vee \neg q \models \neg(p \vee q)$	$p = \text{F}, q = \text{T}$	root $\neg p \vee \neg q, \neg\neg(p \vee q)$; add $p \vee q$; split $\neg p \vee \neg q$; under $\neg p$, $p \vee q$ gives $p \times$ or q : open
(d) $p \rightarrow q, \neg p \rightarrow (r \vee q), \neg r \models p$	$p = \text{F}, q = \text{T}, r = \text{F}$	$p \rightarrow q$: $\neg p$ or q ; under $\neg p$, $\neg p \rightarrow (r \vee q)$ gives $\neg\neg p \times$ or $r \vee q$; then $r \times$ or q : open
(f) $(p \rightarrow q) \rightarrow r, \neg r \models q$	$p = \text{T}, q = \text{F}, r = \text{F}$	$\neg(p \rightarrow q)$ or $r \times$; $\neg(p \rightarrow q)$ gives $p, \neg q$: open
(g) $p \vee q, p \models \neg q$	$p = q = \text{T}$	$\neg\neg q$ gives q ; $p \vee q$: p or q , both open

Each row is the row of Section 3.2's break, whose statements about x take these values: in (a) they are $x > 2$ and $x > 0$ at $x = 1$; in (d), $x > 5$, $x > 0$ and $x > 10$ at $x = 1$; in (f), $x > 0$, $x > 2$ and $x > 3$ at $x = 1$; in (g), $x > 1$ and $x > 2$ at $x = 3$. Section 3.2 found these rows by hand; the search finds them without a guess, and would have said "valid" had there been none.

Two questions remain, and they are the next two sections': does a refuting row show that no derivation exists, and does a valid claim always have a derivation? (Sections 4.7 and 4.8.)

Remark (Looking ahead). Section 4.7 proves that a refuting row rules out every derivation. With it, a search that stays open on a subgoal tells a derivation search to stop chasing that subgoal: the move *test the subgoal*.

4.6.7 Exercises

4.6.7.1 A. Check your understanding

Exercise. A1

- (a) Which of these formulas split when used: $p \wedge q$; $\neg(p \wedge q)$; $\neg(p \rightarrow q)$; $p \leftrightarrow q$; $\neg\neg p$?
- (b) An open finished branch carries $p \rightarrow q$ (used), $\neg q$, $\neg p$, r . Which row does it give?
- (c) True or false: if some branch of a finished tree for $\Gamma, \neg\psi$ is closed, then $\Gamma \models \psi$.
- (d) True or false: two finished trees for the same list, built in different orders, can give different verdicts.
- (e) Why does the warning say that a clash reached under a guess refutes the guess, not the claim?

Solution.

- (a) $\neg(p \wedge q)$ and $p \leftrightarrow q$.
- (b) $p = F$, $q = F$, $r = T$, and every other atom F .
- (c) False: every branch must be closed. One closed branch only rules out the rows of that branch.
- (d) False. By [Proposition 4.6.1](#), a closed tree means that no row makes the list true, and an open finished branch means that one does; both cannot hold.
- (e) The guess picked one side of a split. The rows on the other side were never examined, and a refuting row may lie there.

Exercise. A2: Drill

The parts run from writing one rule, through closing a branch and reading off a row, to short searches.

- (a) What does $\neg(p \vee q)$ add?
- (b) What does $p \rightarrow (q \wedge r)$ add?
- (c) What does $\neg(p \leftrightarrow q)$ add?
- (d) What does $\neg\neg(p \wedge q)$ add?
- (e) Is the branch $p \vee q$, $\neg p$, $\neg q$, q closed?
- (f) Is the branch $p \rightarrow q$, $\neg q$, p closed?
- (g) An open finished branch carries $\neg(p \wedge q)$ (used), p , $\neg q$. Which row does it give?
- (h) Search: does $p \vee q \models p$ hold?
- (i) Search: does $p \wedge q \models p \vee r$ hold?
- (j) Is the list $p \leftrightarrow q$, $q \leftrightarrow \neg p$ satisfiable?

- (k) For the list $p \vee q, \neg(r \rightarrow s)$, which formula should be used first, and why?
- (l) “Suppose q is true; then the branch closes; so the claim is valid.” Is this correct?

Solution.

- (a) $\neg p, \neg q$ (one way).
- (b) $\neg p$, or $q \wedge r$ (two ways).
- (c) $p, \neg q$, or $\neg p, q$.
- (d) $p \wedge q$.
- (e) Yes: it holds q and $\neg q$.
- (f) Not yet. Using $p \rightarrow q$ gives $\neg p$, which closes against p , or q , which closes against $\neg q$; so the tree below it closes.
- (g) $p = \text{T}, q = \text{F}$.
- (h) No. Root $p \vee q, \neg p$; split: p closes, and q is open and finished. The row $p = \text{F}, q = \text{T}$ refutes it.
- (i) Yes. Root $p \wedge q, \neg(p \vee r)$; one way: p, q ; one way: $\neg p, \neg r$; closed, so valid.
- (j) No. The first gives p, q or $\neg p, \neg q$. Under each, the second gives $q, \neg p$ or $\neg q, \neg \neg p$, and all four branches close. The last one, $\neg p, \neg q, \neg q, \neg \neg p$, closes at once: $\neg \neg p$ is the negation of $\neg p$.
- (k) $\neg(r \rightarrow s)$: it is one way.
- (l) No: that is a guess. The case with q false must also close.

4.6.7.2 B. Practice

Exercise. B1: Valid or refuted?

Decide each claim by a search tree; you may stop at the first finished open branch. For a refuted claim, read off a refuting row and check it.

- (a) $p \rightarrow q, r \rightarrow s, p \vee r \models q \vee s$
- (b) $p \rightarrow q, r \rightarrow s, q \vee s \models p \vee r$
- (c) $(p \wedge q) \rightarrow r \models (p \rightarrow r) \vee (q \rightarrow r)$
- (d) $p \leftrightarrow q, q \vee r \models p \vee \neg r$

Solution. The verdict comes first, then the tree as its stages; $\times$ marks a side that closes.

- (a) **Valid.** $\neg(q \vee s)$ adds $\neg q, \neg s$. $p \vee r$: p or r . Under p : $p \rightarrow q$ gives $\neg p \times$ or $q \times$. Under r : $r \rightarrow s$ gives $\neg r \times$ or $s \times$. Every branch is closed.
- (b) **Refuted.** $\neg(p \vee r)$ adds $\neg p, \neg r$. $q \vee s$: q or s . Under q : $p \rightarrow q$ gives $\neg p$ or q ; take $\neg p$. Then $r \rightarrow s$ gives $\neg r$ or s ; take $\neg r$. The branch through $q, \neg p, \neg r$ is open and finished, and the search stops there; the other sides are not followed. Row $p = \text{F}, q = \text{T}, r = \text{F}, s = \text{F}$: both implications have false hypotheses, $q \vee s$ is T , and $p \vee r$ is F .
- (c) **Valid.** $\neg((p \rightarrow r) \vee (q \rightarrow r))$ adds $\neg(p \rightarrow r), \neg(q \rightarrow r)$; these add $p, \neg r$ and $q, \neg r$. $(p \wedge q) \rightarrow r$: $\neg(p \wedge q)$ or $r \times$. $\neg(p \wedge q)$: $\neg p \times$ or $\neg q \times$.
- (d) **Refuted.** $\neg(p \vee \neg r)$ adds $\neg p, \neg \neg r$, and $\neg \neg r$ adds r . $p \leftrightarrow q$: $p, q \times$ (against $\neg p$) or

$\neg p, \neg q, q \vee r$: $q \times$ or r . The branch $\neg p, \neg q, r$ is open and finished. Row $p = F, q = F, r = T$: $p \leftrightarrow q$ is T, $q \vee r$ is T, and $p \vee \neg r$ is F.

Claims (a) and (b) are look-alikes, with the premises' disjunction and the conclusion traded, and only (a) holds.

Exercise. B2: Can all five be true?

Let n be an integer, and consider:

- (1) if n is divisible by 6, then n is divisible by 4;
 - (2) n is divisible by 6, or n is odd;
 - (3) n is not odd;
 - (4) if n is divisible by 4, then n is not divisible by 3;
 - (5) if n is divisible by 6, then n is divisible by 3.
- (a) With the atoms s, f, o, t for “divisible by 6”, “divisible by 4”, “odd”, “divisible by 3”, decide by a search tree whether the list of five formulas is satisfiable. (b) Drop one claim so that the rest are satisfiable, and give an integer for which they all hold.

Solution.

- (a) The list is $s \rightarrow f, s \vee o, \neg o, f \rightarrow \neg t, s \rightarrow t$. No rule is one way. Split $s \vee o$, which has a side that closes at once: s , or $o \times$ (against $\neg o$). Under s : $s \rightarrow f$ gives $\neg s \times$ or f ; $f \rightarrow \neg t$ gives $\neg f \times$ or $\neg t$; $s \rightarrow t$ gives $\neg s \times$ or $t \times$ (against $\neg t$). Every branch closes: the list is unsatisfiable, so no integer makes all five claims true, and their form alone forbids it.
- (b) Drop (4). In the tree without it, the branch s, f, t , with o false, is open and finished. The integer $n = 12$ is divisible by 6, 4 and 3, and not odd, so it satisfies (1), (2), (3) and (5). Dropping (1) instead also works: under s , the rule of $f \rightarrow \neg t$ gives $\neg f$, then $s \rightarrow t$ gives t , and the branch $s, \neg f, t$, with o false, is open and finished; $n = 6$ realizes it.

Marking scheme. (6 points)

- 1: the formulas.
- 3: the closed tree.
- 2: (b), with an integer checked against each remaining claim.

Exercise. B3: Name the move, then complete it

A search for $p \vee (q \wedge r) \models p \vee q$ has reached this stage.

$$\begin{array}{c}
 p \vee (q \wedge r) \\
 \neg(p \vee q) \quad \checkmark \\
 \neg p \\
 \neg q \\
 | \\
 ?
 \end{array}$$

Name the move, then complete it.

Solution. No one-way formula is left unused, so the move is to split the disjunction $p \vee (q \wedge r)$. The side p closes against $\neg p$. The side $q \wedge r$ is one way: it adds q, r , and closes against $\neg q$. Every branch is closed: the entailment holds.

$$\begin{array}{c}
 p \vee (q \wedge r) \quad \checkmark \\
 \neg(p \vee q) \quad \checkmark \\
 \neg p \\
 \neg q \\
 \swarrow \quad \searrow \\
 \begin{array}{cc}
 p & q \wedge r \quad \checkmark \\
 \times & q \\
 & r \\
 & \times
 \end{array}
 \end{array}$$

Marking scheme. (4 points)

- 1: the move named (split, since nothing one-way is left).
- 2: the completed tree.
- 1: the verdict.

4.6.7.3 C. Going deeper

Exercise. C1: Rules for a new connective

Add the connective $|$ (“not both”), with the clause (V7) of Exercise (C1: Substitution with a new connective, the last half blank, used later): $\varphi | \psi$ is F exactly when both parts are T.

- Read search rules for $\varphi | \psi$ and for $\neg(\varphi | \psi)$ off (V7), and say which is one way.
- Decide $p | q$, $p \models \neg q$ by a search.
- Decide by a search whether $p | (p | p)$ is a tautology.

Solution.

- $\varphi | \psi$ is true unless both parts are true, so it is true in two ways: it adds $\neg\varphi$ or $\neg\psi$. $\neg(\varphi | \psi)$ is true exactly when both parts are true, one way: it adds φ, ψ .
- Root $p | q$, p , $\neg\neg q$. The formula $\neg\neg q$ adds q . Then $p | q$ gives $\neg p \times$ or $\neg q \times$. Every branch is closed: valid.
- The root is the negated formula, $\neg(p | (p | p))$. It is one way: it adds p and $p | p$. The latter gives $\neg p \times$ or $\neg p \times$. The tree is closed, so the negation is unsatisfiable and the

formula is a tautology. It says “not both p and not p ”, since $p \mid p$ is true exactly when p is false.

Marking scheme. (8 points)

- 3: (a), with each rule tied to its rows of (V7).
- 2: (b).
- 3: (c), including the root as the negated formula.

Exercise. C2: Exit check: refuting-row search

Close the section first. For each claim, run a search tree, state the tactics you used in your order of moves, give the verdict, and say which part of [Proposition 4.6.1](#) justifies it.

- (a) $p \rightarrow (q \vee r), \neg r \models p \rightarrow q$
 (b) $p \vee q, p \rightarrow r \models q \vee \neg r$

Solution.

- (a) The conclusion first: $\neg(p \rightarrow q)$ adds $p, \neg q$. Then $p \rightarrow (q \vee r)$, which has a side that closes at once: $\neg p \times$ or $q \vee r$. Then $q \vee r$: $q \times$, or $r \times$ (against $\neg r$). Every branch is closed, so by (b) the counterexample set is unsatisfiable, and the entailment holds.
- (b) The conclusion first: $\neg(q \vee \neg r)$ adds $\neg q, \neg \neg r$, and then $\neg \neg r$ adds r (one way before two ways). Split $p \vee q$, whose side q closes at once: p , or $q \times$. Under p : $p \rightarrow r$ gives $\neg p \times$ or r , and the branch is open and finished (stop at the first finished open branch). By (c), its row $p = T, q = F, r = T$ makes every root formula true: $p \vee q$ is T, $p \rightarrow r$ is T, and $q \vee \neg r$ is F. The claim is refuted.

Marking scheme. (10 points)

- 1 + 1: the roots, which are the counterexample sets.
- 3 + 3: the trees.
- 1: the tactics named.
- 1: the parts of [Proposition 4.6.1](#) cited correctly, (b) for (a) and (c) for (b).

Fatal errors (0 for the part): a root without the negated conclusion; a verdict from an unfinished open branch.

Exercise. C3: Challenge: the search with one more connective

Extend search trees by the rules of C1 for $\mid$. Prove that [Proposition 4.6.1](#) still holds for lists of formulas of the extended language: give the new cases of each part of its proof.

Solution.

- (a) The new rules add immediate parts or their negations, so every added formula is still a subformula of a root formula or the negation of one, and the count of the proof is unchanged.
- (b) New cases of the claim. If v makes $\varphi \mid \psi$ true, then by (V7) not both parts are true,

so v makes $\neg\varphi$ or $\neg\psi$ true: one of the two new nodes. If v makes $\neg(\varphi \mid \psi)$ true, then $\varphi \mid \psi$ is false, so both parts are true: the one new node.

- (c) New cases of the induction. A used $\varphi \mid \psi$ on B has $\neg\varphi$ or $\neg\psi$ on B , which is shorter and true by the hypothesis, so one part of $\varphi \mid \psi$ is false and $\varphi \mid \psi$ is true by (V7). A used $\neg(\varphi \mid \psi)$ has φ and ψ on B , both shorter and true by the hypothesis, so $\varphi \mid \psi$ is false and its negation true. The base cases are unchanged: an atom, a negated atom, $\perp$ and $\neg\perp$.

Marking scheme. (10 points)

- 2: (a).
- 4: (b), both rules.
- 4: (c), both rules, with the hypothesis applied to shorter formulas.

Fatal error: in (c), using the hypothesis for $\varphi \mid \psi$ itself.