Automated Theorem Proving 1/4: Introduction and Propositional Theorem Proving

A.L. Lamprecht

Course Program Semantics and Verfication 2020, Utrecht University

September 21, 2020

Lecture Notes

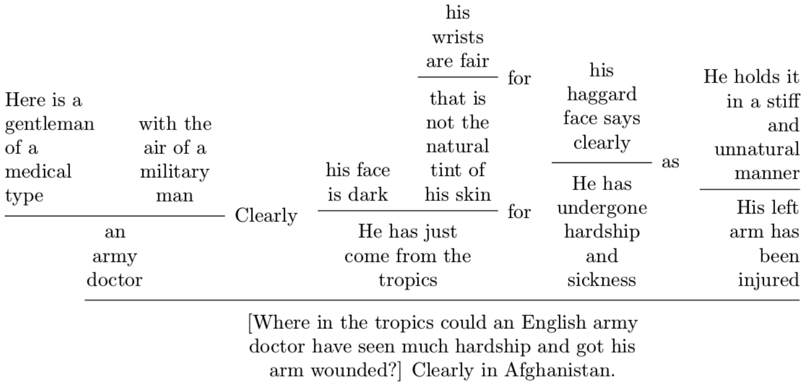
"Automated Reasoning" by Gerard A.W. Vreeswijk.
Available for download on the course website.
My slides are largely based on them.

What is Reasoning?

"... You appeared to be surprised when I told you, on our first meeting, that you had come from Afghanistan."
"You were told, no doubt."
"Nothing of the sort. I knew you came from Afghanistan. From long habit the train of thought ran so swiftly through my mind that I arrived at the conclusion without being conscious of intermediate steps. There were such steps, however. The train of reasoning ran. 'Here is a gentleman of a medical type, but with the air of a military man. Clearly an army doctor, then. He has just come from the tropics, for his face is dark, and that is not the natural tint of his skin, for his wrists are fair. He has undergone hardship and sickness, as his haggard face says clearly. His left arm has been injured. He holds it in a stiff and unnatural manner. Where in the tropics could an English army doctor have seen much hardship and got his arm wounded? Clearly in Afghanistan.' The whole train of thought did not occupy a second. I then remarked that you came from Afghanistan, and you were astonished."
"It is simple enough as you explain it," I said, smiling.

(from: *A Study in Scarlet*, Sir Arthur Conan Doyle)

Formalization of Reasoning



Here is a gentleman of a medical type

with the air of a military man

an army doctor

Clearly

his wrists are fair

for

that is not the natural tint of his skin

his face is dark

for

He has just come from the tropics

his haggard face says clearly

as

He has undergone hardship and sickness

He holds it in a stiff and unnatural manner

His left arm has been injured

[Where in the tropics could an English army doctor have seen much hardship and got his arm wounded?] Clearly in Afghanistan.

Figure 1.4: A "train of reasoning" (Sir Arthur Conan Doyle)

Formalization of Reasoning

$$\frac{\dfrac{A \quad B}{C} \quad \text{Clearly} \quad \dfrac{D}{E} \text{ for } \dfrac{\dfrac{F}{G} \text{ for}}{H} \text{ for } \dfrac{I}{J} \text{ as } \dfrac{K}{L}}{M}$$

Figure 1.5: The same “train of reasoning,” but now symbolized

Automated Reasoning - Brief History

- Reasoning exists as long as mankind, and the desire to mechanize reasoning is very old.
- Gottfried Wilhelm Leibniz (1646-1716) articulates ideas of merging calculation and reasoning.
- From 1945: With the rise of the computer automated theorem proving became a dedicated research area.
- Robinson (1965): resolution proof calculus
- Generations of theorem provers: P1, RW1, Otter, Prover9

Applications of Automated Reasoning

- Program verfication
- Hardware verification
- Error diagnosis and explanation
- Identification of modules that match a specification
- Planning and scheduling
- Knowledge integration for natural-language understanding
- Mathematical theorem proving
- Weather forecasting
- Tide monitoring
- Legal reasoning
- Argumentation and negotiation
- ...

Kinds of Automated Reasoning

- Automated deduction
  - Symbolic approaches
    - **Semantic tableaux**
    - **Resolution**
    - Binary decision diagrams
  - Connectionistic approaches
- Non-deductive forms of automated reasoning
  - Symbolic (or qualitative) approaches
    - Argumentation-based approaches
    - Qualitative probabilistic networks
    - Non-monotonic reasoning
  - Numeric (or quantitative) approaches
    - Probabilistic or probabilistically oriented approaches
    - Fuzzy logics
    - Possibilistic approaches
  - Connectionistic and/or holistic approaches
    - Inference systems based on coherence
  - Hybrid approaches

In This Course

- Propositional theorem proving (today),
  Chapter 2 of the lecture notes
- First-order theorem proving (Wednesday),
  Chapter 3 of the lecture notes
- Clause sets and resolution (next Monday),
  Chapters 4 and 5 of the lecture notes
- Satisfiability checkers, SAT/SMT (next Wednesday),
  Chapter 6 of the lecture notes, additional material

Propositional Theorem Proving

- Remember your first course in logic?
- Remember doing proofs of obscure formulas such as $p \supset (q \supset p)$?
- The good news: Automating propositional theorem proving is straightforward.
- We'll get to the bad news later.

The Nature of Theorem Proving

One possible way to prove theorems:

- Begin with axioms and rules of inference.
- Infer theorems on the basis of the axioms and theorems that were inferred earlier.

Example: Hilbert's system (three axioms and one rule of inference)

Reminder: Hilbert's System

Axioms:

- $\phi \supset (\psi \supset \phi)$
- $(\phi \supset (\psi \supset \chi)) \supset ((\phi \supset \psi) \supset (\phi \supset \chi))$
- $(\neg\phi \supset \neg\psi) \supset (\psi \supset \phi)$

where ϕ, ψ and χ may be any propositional formula.

Rule of inference:

$$\frac{\phi \quad \phi \supset \psi}{\psi} \text{ \textit{modus ponens}}$$

where ϕ and ψ may be any propositional formula.

Exercise: Prove $p \supset q, q \supset r \vdash p \supset r$

Use Hilbert's system to prove

$$p \supset q, q \supset r \vdash p \supset r$$

Solution: Proof of $p \supset q, q \supset r \vdash p \supset r$

1. $p \supset q$	(Hypothesis)
2. $q \supset r$	(Hypothesis)
3. $(q \supset r) \supset (p \supset (q \supset r))$	(Instance of Axiom 1)
4. $(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$	(Instance of Axiom 2)
5. $p \supset (q \supset r)$	(From 2, 3 by MP)
6. $(p \supset q) \supset (p \supset r)$	(From 4, 5 by MP)
7. $p \supset r$	(From 1, 6 by MP)

Process

How did you proceed? Trying out things until it worked?

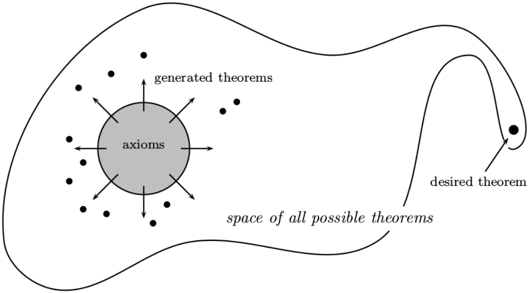


Figure 2.1: The British Museum Algorithm

Not effective. Most "real" theorem proving is done by refutation.

Searching for a Counterexample

Suppose we want to investigate the validity of a sequent in propositional logic, for example:

$$\neg q, \neg(p \wedge q) \vdash p \supset q$$

A systematic way to determine its status, is to try to make it false. If we have considered every possibility to falsify it and fail, then we have proven that is is valid.

Let's try.

Searching for a Counterexample

To falsify $\neg q, \neg(p \wedge q) \vdash p \supset q$, we have to make both of $\neg q, \neg(p \wedge q)$ true and $p \supset q$ false. Write as:

$$\text{TRUE: } \neg q, \neg(p \wedge q) \ ; \quad \text{FALSE: } p \supset q$$

Searching for a Counterexample

To falsify $\neg q, \neg(p \wedge q) \vdash p \supset q$, we have to make both of $\neg q, \neg(p \wedge q)$ true and $p \supset q$ false. Write as:

$$\text{TRUE: } \neg q, \neg(p \wedge q) \ ; \quad \text{FALSE: } p \supset q$$

Make p true and q false to make the implication on the RHS false:

$$\text{TRUE: } \neg q, \neg(p \wedge q), p \ ; \quad \text{FALSE: } q$$

Searching for a Counterexample

To falsify $\neg q, \neg(p \wedge q) \vdash p \supset q$, we have to make both of $\neg q, \neg(p \wedge q)$ true and $p \supset q$ false. Write as:

$$\text{TRUE: } \neg q, \neg(p \wedge q) \ ; \quad \text{FALSE: } p \supset q$$

Make p true and q false to make the implication on the RHS false:

$$\text{TRUE: } \neg q, \neg(p \wedge q), p \ ; \quad \text{FALSE: } q$$

Make q false to make $\neg q$ true:

$$\text{TRUE: } \neg(p \wedge q), p \ ; \quad \text{FALSE: } q, q$$

(continue on next slide)

Searching for a Counterexample

TRUE: $\neg(p \wedge q), p$; FALSE: q, q

Searching for a Counterexample

$$\text{TRUE: } \neg(p \wedge q), p \;; \quad \text{FALSE: } q, q$$

Make $p \wedge q$ false to make $\neg(p \wedge q)$ true:

$$\text{TRUE: } p \;; \quad \text{FALSE: } q, q, p \wedge q$$

Searching for a Counterexample

$$\text{TRUE: } \neg(p \wedge q), p \ ; \quad \text{FALSE: } q, q$$

Make $p \wedge q$ false to make $\neg(p \wedge q)$ true:

$$\text{TRUE: } p \ ; \quad \text{FALSE: } q, q, p \wedge q$$

Now two directions – either falsify p or q:

$(1)\ \text{TRUE: } p \ ; \ \ \text{FALSE: } q, q, p \qquad (2)\ \text{TRUE: } p \ ; \ \ \text{FALSE: } q, q, q$

Searching for a Counterexample

$$\text{TRUE: } \neg(p \wedge q), p \;;\quad \text{FALSE: } q, q$$

Make $p \wedge q$ false to make $\neg(p \wedge q)$ true:

$$\text{TRUE: } p \;;\quad \text{FALSE: } q, q, p \wedge q$$

Now two directions – either falsify p or q:

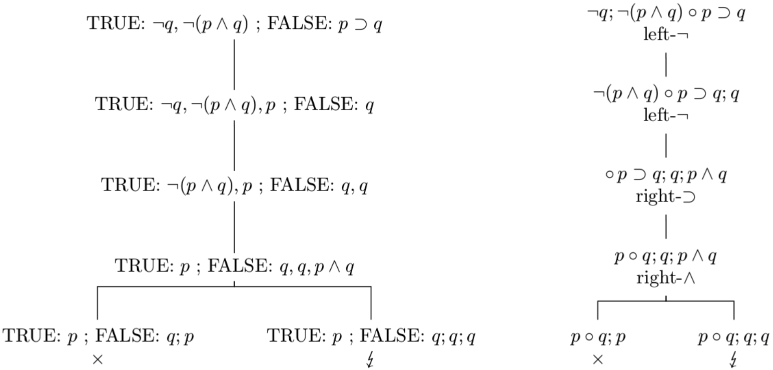
(1) TRUE: p ; FALSE: q, q, p (2) TRUE: p ; FALSE: q, q, q

All formulas are atomic now, so we can check for counterexamples. (1) does not contain a counterexample, because it is impossible to make p both true and false.

(2) does contain a counterexample: $\{p = 1,\ q = 0\}$.

Hence, $\neg q, \neg(p \wedge q) \vdash p \supset q$ is invalid.

Semantic Tableaux (aka Refutation Trees, Semantic Trees)



TRUE: $\neg q, \neg(p \wedge q)$; FALSE: $p \supset q$

TRUE: $\neg q, \neg(p \wedge q), p$; FALSE: q

TRUE: $\neg(p \wedge q), p$; FALSE: q, q

TRUE: p ; FALSE: $q, q, p \wedge q$

TRUE: p ; FALSE: $q; p$

×

TRUE: p ; FALSE: $q; q; q$

↯

$\neg q; \neg(p \wedge q) \circ p \supset q$

left-$\neg$

$\neg(p \wedge q) \circ p \supset q; q$

left-$\neg$

$\circ p \supset q; q; p \wedge q$

right-$\supset$

$p \circ q; q; p \wedge q$

right-$\wedge$

$p \circ q; p$

×

$p \circ q; q; q$

↯

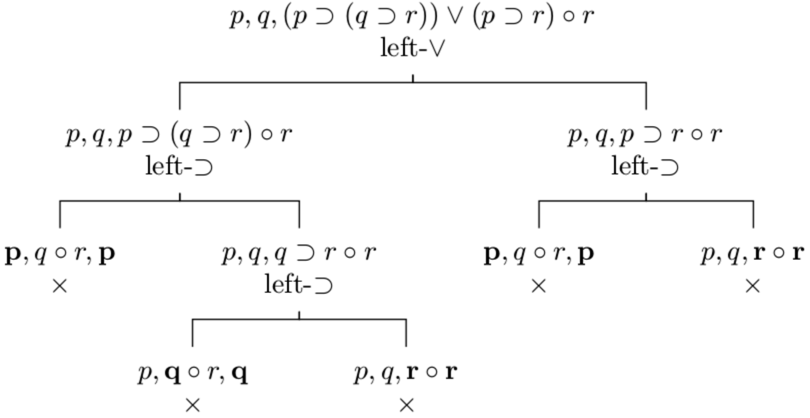
Table 2.1: Refutation trees of $\neg q, \neg(p \wedge q) \vdash p \supset q$.

Terminology

We say that a branch is ...

- **closed** if the search for a counterexample has terminated for this particular branch, i.e. two equal atoms occur on both sides of the $\circ$,
- **open** if the two sides of the $\circ$ do not share a common atom,
- **complete** if it cannot be further extended (closed or all formulas on it have been analyzed),
- **saturated** if the branch is complete but open.

Failing Refutation



$p, q, (p \supset (q \supset r)) \vee (p \supset r) \circ r$

left-$\vee$

$p, q, p \supset (q \supset r) \circ r$

left-$\supset$

$p, q, p \supset r \circ r$

left-$\supset$

$\mathbf{p}, q \circ r, \mathbf{p}$

$\times$

$p, q, q \supset r \circ r$

left-$\supset$

$\mathbf{p}, q \circ r, \mathbf{p}$

$\times$

$p, q, \mathbf{r} \circ \mathbf{r}$

$\times$

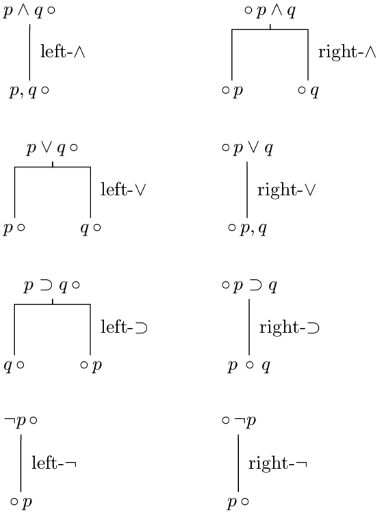
$p, \mathbf{q} \circ r, \mathbf{q}$

$\times$

$p, q, \mathbf{r} \circ \mathbf{r}$

$\times$

Analytic Refutation Rules



$p \wedge q \circ$

left-$\wedge$

$p, q \circ$

$\circ\, p \wedge q$

right-$\wedge$

$\circ\, p$ $\quad$ $\circ\, q$

$p \vee q \circ$

left-$\vee$

$p \circ$ $\quad$ $q \circ$

$\circ\, p \vee q$

right-$\vee$

$\circ\, p, q$

$p \supset q \circ$

left-$\supset$

$q \circ$ $\quad$ $\circ\, p$

$\circ\, p \supset q$

right-$\supset$

$p \circ q$

$\neg p \circ$

left-$\neg$

$\circ\, p$

$\circ\, \neg p$

right-$\neg$

$p \circ$

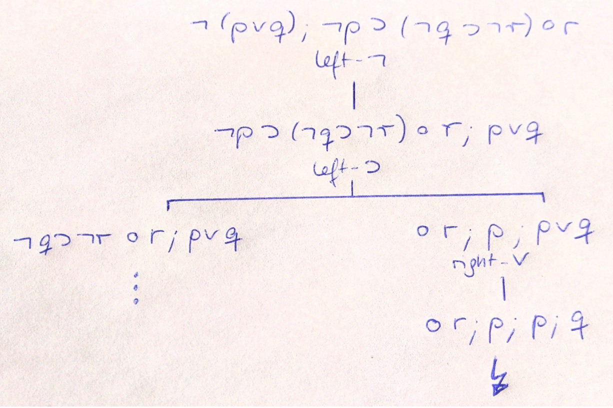
Exercise

Problem 3 (a) from Section 2.2 in the Lecture Notes: Construct a refutation tree for the following sequent. Specify a counterexample if the sequent turns out to be invalid.

$$\neg(p \vee q), \neg p \supset (\neg q \supset \neg r) \vdash r$$

Solution

Invalid sequent. Countermodel: w, with $w(p) = w(q) = w(r) = 0$.


$\neg(p \vee q); \neg p \supset (\neg q \supset \neg r) \circ r$
left-$\neg$
$\neg p \supset (\neg q \supset \neg r) \circ r; p \vee q$
left-$\supset$
$\neg q \supset \neg r \circ r; p \vee q$
$\vdots$
$\circ r; p; p \vee q$
right-$\vee$
$\circ r; p; p; q$
↯

Analytic Refutation Rules

Three crucial properties:

1. *Sub-formula property.* Each rule analyzes a formula on the basis of its outermost connective.
2. *Complete analysis.* Each non-atomic formula in a sequent can be analyzed by at least onerule.
3. *Unique analysis.* Each non-atomic formula in a sequent can be analyzed by at most one rule.

Turning a Refutation Tree into a Proof

- If all branches in a refutation tree are closed, and no branch ends in a counterexample, it may be concluded that the refutation has failed.
- We may thus consider the tree as a *proof* of the sequent in question.
- To turn a refutation tree into proof, we turn it upside down and supply every step with a justification.
- We obtain a so-called *cut-free* proof in the *Gentzen sequent calculus*.

Example: Proof for $p, q, ((p \supset (q \supset r)) \vee (p \supset r)) \Rightarrow r$

$$\dfrac{\dfrac{p, q \Rightarrow r, p \qquad \dfrac{p, q \Rightarrow r, q \quad p, q, r \Rightarrow r}{p, q, (q \supset r) \Rightarrow r}\ \text{left-}\supset}{p, q, (p \supset (q \supset r)) \Rightarrow r}\ \text{left-}\supset \qquad \dfrac{p, q \Rightarrow r, p \quad p, q, r \Rightarrow r}{p, q, (p \supset r) \Rightarrow r}\ \text{left-}\supset}{p, q, ((p \supset (q \supset r)) \vee (p \supset r)) \Rightarrow r}\ \text{left-}\vee$$

Gentzen System for Propositional Logic

left-$\wedge$:

$$\frac{\ldots, p, q, \ldots \vdash \ldots}{\ldots, p \wedge q, \ldots \vdash \ldots}$$

left-$\vee$:

$$\frac{\ldots, p, \ldots \vdash \ldots \qquad \ldots, q, \ldots \vdash \ldots}{\ldots, p \vee q, \ldots \vdash \ldots}$$

left-$\supset$:

$$\frac{\ldots \vdash \ldots, p, \ldots \qquad \ldots q \ldots \vdash \ldots}{\ldots, p \supset q, \ldots \vdash \ldots}$$

left-$\neg$:

$$\frac{\ldots \vdash \ldots, p, \ldots}{\ldots, \neg p, \ldots \vdash \ldots}$$

right-$\wedge$:

$$\frac{\ldots \vdash \ldots, p, \ldots \qquad \ldots \vdash \ldots, q, \ldots}{\ldots \vdash \ldots, p \wedge q, \ldots}$$

right-$\vee$:

$$\frac{\ldots \vdash \ldots, p, q, \ldots}{\ldots \vdash \ldots, p \vee q, \ldots}$$

right-$\supset$:

$$\frac{\ldots, p, \ldots \vdash \ldots, q, \ldots}{\ldots \vdash \ldots, p \supset q, \ldots}$$

right-$\neg$:

$$\frac{\ldots, p, \ldots \vdash \ldots}{\ldots \vdash \ldots, \neg p, \ldots}$$

Sound- and Completeness of Refutation Trees

Two questions remain to be answered:

1. *Soundness* - we must be certain that refutations fail **only** for valid sequences.
2. *Completeness* - we must be certain that refutations fail for **all** valid sequences.

The good news is: Refutation trees are sound and complete. (See the lecture notes for the respective proofs.)

A Hard Case

- Analytic refutation is conceptually simple.
- However, sometimes the trees grow extremely fast.
- Reason: the method explores parts of the search space that have already been explored.
- This can lead to bad performance - sometimes even a brute-force truth table approach is faster!
- Something better is needed...

The Bad News: Propositional Theorem Proving is NP-Complete

- Propositional ATP amounts to finding a countermodel for a propositional formula.
- This problem is known as the *satisfiability problem*, or *SAT*.
- SAT is NP-complete.
- Thus, it is extremely likely (in fact, almost certain) that no efficient algorithm for propositional ATP exists.

What to do?

- Use an exponential solution anyway. (At least it will solve the problem exactly.)
- Use a heuristic. (Next in the Lecture Notes, but not discussed in the lecture.)
- Solve the problem approximately instead of exactly. (Part II of the Lecture Notes, not discussed in the lecture.)
- Choose a better abstraction. (Not discussed.)

In This Course

- Propositional theorem proving (today), Chapter 2 of the lecture notes
- First-order theorem proving (Wednesday), Chapter 3 of the lecture notes
- Clause sets and resolution (next Monday), Chapters 4 and 5 of the lecture notes
- Satisfiability checkers, SAT/SMT (next Wednesday), Chapter 6 of the lecture notes, additional material

Homework (1)

The following exercises are useful to review today's content in preparation for the next lecture:

- Sec. 2.2 Problem 1 (page 28 in the lecture notes)
- Sec. 2.2 Problem 3 (b)-(e) (page 29)
- Sec. 2.3 Problem 1 (page 32)
- Sec. 2.4 Problems 1 and 2 (pages 35/36)

Homework (2)

The next lecture assumes familiarity with the syntax and semantics of first-order logics. In particular:

- well-formed formulas,
- scope of a quantifier,
- free and bound variables,
- closed well-formed formulas (sentences),
- fair substitutions,
- first-order domains,
- interpretation of constants, function symbols and predicate symbols,
- interpretation of well-formed formulas,
- variable assignments,
- first-order models,
- first-order countermodels

Sounds familiar? If not, exercises 2–8 in Section 3.1 of the lecture notes will help you to brush up your knowledge of the syntax and semantics of first-order logic.