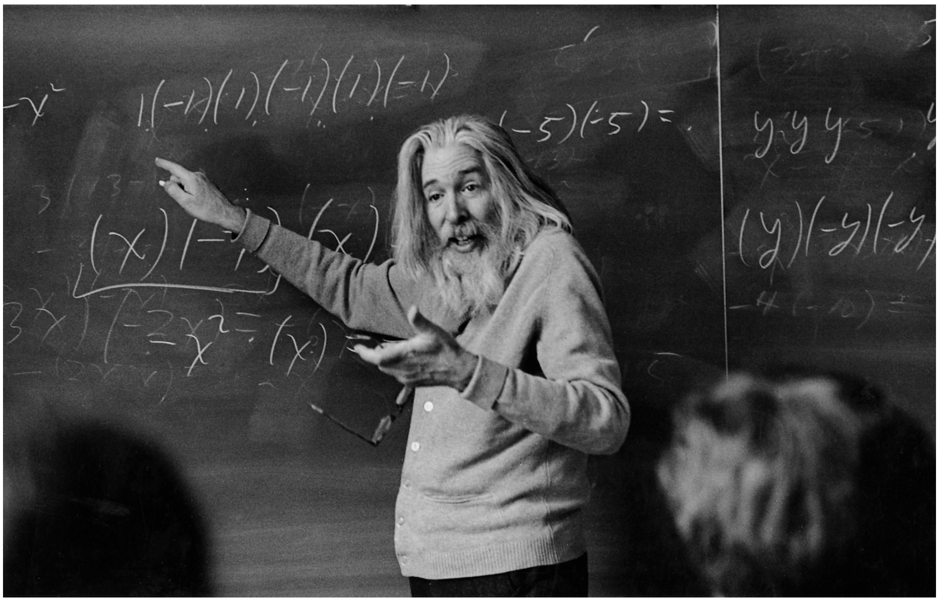
Raymond Smullyan's Tableaux Propositional Logic



Courtesy of Chiara Ghedini (FBK, Trento)

Outline of this lecture

- An introduction to Automated Reasoning with Analytic Tableaux;
- Today we will be looking into tableau methods for classical propositional logic (well discuss first-order tableaux later).
- Analytic Tableaux are a a family of mechanical proof methods, developed for a variety of different logics. Tableaux are nice, because they are both easy to grasp for *humans* and easy to implement on *machines*.

How does it work?

The tableau method is a method for proving, in a mechanical manner, that a given set of formulas is not satisfiable. In particular, this allows us to perform automated *deduction*:

Given :	set of premises Γ and conclusion ϕ
Task :	prove $\Gamma \models \phi$
How?	show $\Gamma \cup \neg\phi$ is not satisfiable (which is equivalent),
	i.e. add the complement of the conclusion to the premises
	and derive a contradiction (refutation procedure)

Reduce Logical Consequence to (un)Satisfiability

Theorem

$\Gamma \models \phi$ *if and only if* $\Gamma \cup \{\neg\phi\}$ *is unsatisfiable*

Proof.

$\Rightarrow$ Suppose that $\Gamma \models \phi$, this means that every interpretation $\mathcal{I}$ that satisfies Γ, it does satisfy ϕ, and therefore $\mathcal{I} \not\models \neg\phi$. This implies that there is no interpretations that satisfies together Γ and $\neg\phi$.

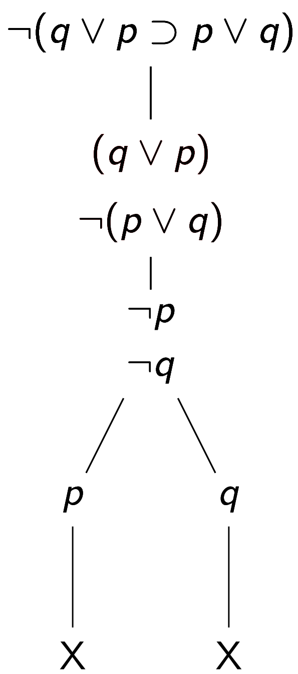
$\Leftarrow$ Suppose that $\mathcal{I} \models \Gamma$, let us prove that $\mathcal{I} \models \phi$, Since $\Gamma \cup \{\neg\phi\}$ is not satisfiable, then $\mathcal{I} \not\models \neg\phi$ and therefore $\mathcal{I} \models \phi$.

$\square$

Constructing Tableau Proofs

- **Data structure**: a proof is represented as a tableaua binary tree, the nodes of which are labelled with formulas.
- **Start**: we start by putting the premises and the negated conclusion into the root of an otherwise empty tableau.
- **Expansion**: we apply expansion rules to the formulas on the tree, thereby adding new formulas and splitting branches.
- **Closure**: we close branches that are obviously contradictory.
- **Success**: a proof is successful iff we can close all branches.

An example



Expansion Rules of Propositional Tableau

α rules

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi\\ \psi\end{array}} \qquad \frac{\neg(\phi \vee \psi)}{\begin{array}{c}\neg\phi\\ \neg\psi\end{array}} \qquad \frac{\neg(\phi \supset \psi)}{\begin{array}{c}\phi\\ \neg\psi\end{array}}$$

$\neg\neg$-Elimination

$$\frac{\neg\neg\phi}{\phi}$$

β rules

$$\frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\neg(\phi \wedge \psi)}{\neg\phi \mid \neg\psi} \qquad \frac{\phi \supset \psi}{\neg\phi \mid \psi}$$

Branch Closure

$$\frac{\begin{array}{c}\phi\\ \neg\phi\end{array}}{X}$$

Note: These are the standard ("Smullyan-style") tableau rules.

We omit the rules for $\equiv$. We rewrite $\phi \equiv \psi$ as $(\phi \supset \psi) \wedge (\psi \supset \phi)$

Smullyans Uniform Notation

Two types of formulas: conjunctive (α) and disjunctive (β):

α	α_1	α_2
$\phi \wedge \psi$	ϕ	ψ
$\neg(\phi \vee \psi)$	$\neg\phi$	$\neg\psi$
$\neg(\phi \supset \psi)$	ϕ	$\neg\psi$

β	β_1	β_2
$\phi \vee \psi$	ϕ	ψ
$\neg(\phi \wedge \psi)$	$\neg\phi$	$\neg\psi$
$\phi \supset \psi$	$\neg\phi$	ψ

We can now state α and β rules as follows:

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Note: α rules are also called deterministic rules. β rules are also called splitting rules.

Some definition for tableaux

Definition (Closed branch)

A closed branch is a branch which contains a formula and its negation.

Definition (Open branch)

An open branch is a branch which is not closed

Definition (Closed tableaux)

A tableaux is closed if all its branches are closed.

Definition

Let ϕ and Γ be a propositional formula and a finite set of propositional formulae, respectively. We write $\Gamma \vdash \phi$ to say that there exists a closed tableau for $\Gamma \cup \{\neg\phi\}$

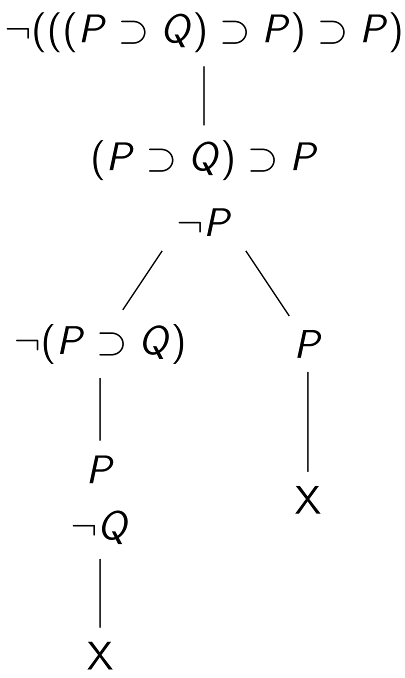
Exercises

Exercise

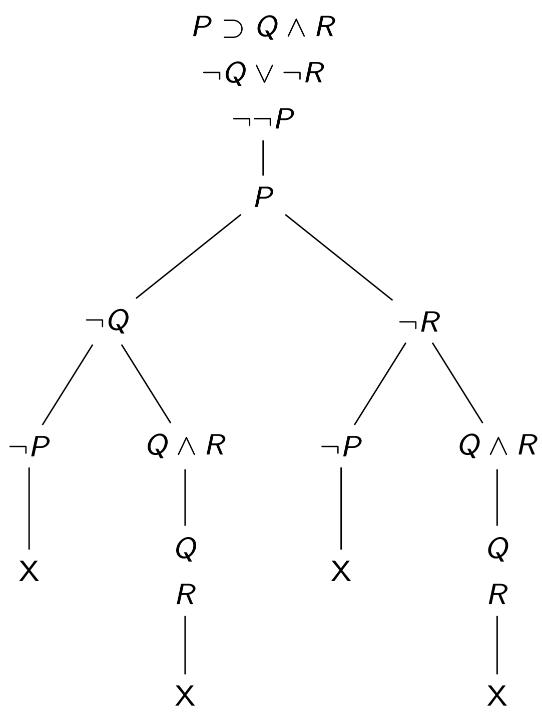
Show that the following are valid arguments:

- $\models ((P \supset Q) \supset P) \supset P$
- $P \supset (Q \wedge R), \neg Q \vee \neg R \models \neg P$

Solutions



Solutions



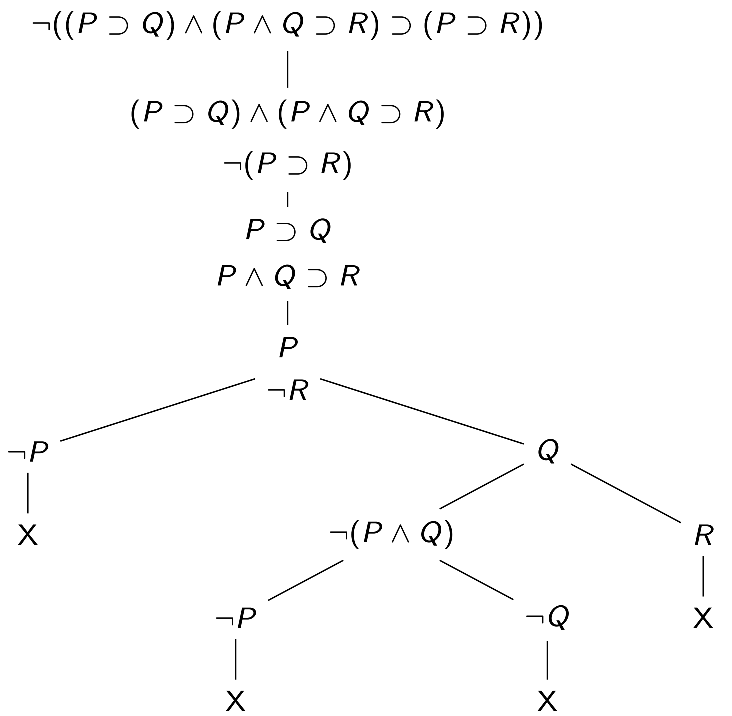
Note: different orderings of expansion rules are possible! But all lead to unsatisfiability.

Exercises

Exercise

Check whether the formula $\neg((P \supset Q) \wedge (P \wedge Q \supset R) \supset (P \supset R))$ is satisfiable

Solution



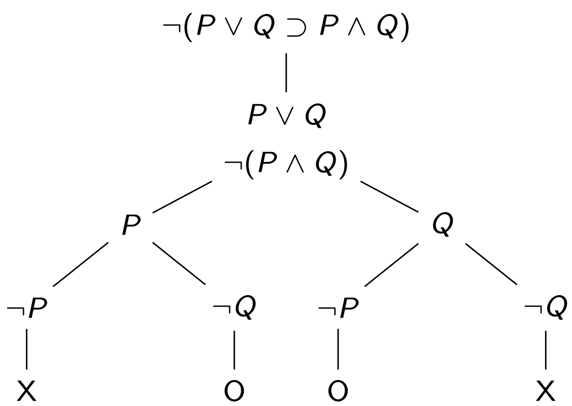
The tableau is closed and the formula is not satisfiable.

Satisfiability: An example

Exercise

Check whether the formula $\neg(P \vee Q \supset P \wedge Q)$ is satisfiable

Solution



Two open branches. The formula is satisfiable.

The tableau shows us all the possible interpretations $(\{P\}, \{Q\})$ that satisfy the formula.

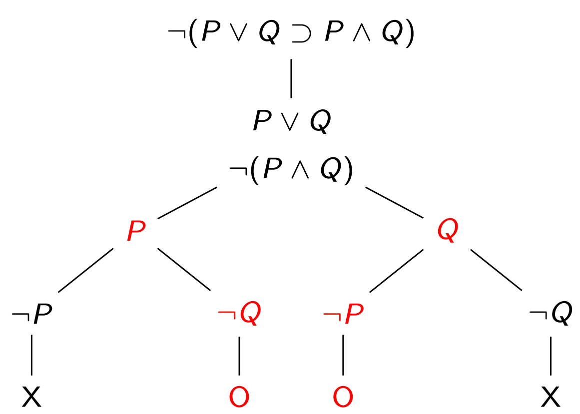
Using the tableau to build interpretations.

For each open branch in the tableau, and for each propositional atom p in the formula we define

$$\mathcal{I}(p) = \begin{cases} \text{True} & \text{if } p \text{ belongs to the branch,} \\ \text{False} & \text{if } \neg p \text{ belongs to the branch.} \end{cases}$$

If neither p nor $\neg p$ belong to the branch we can define $\mathcal{I}(p)$ in an arbitrary way.

Models for $\neg(P \vee Q \supset P \wedge Q)$



Two models:

- $\mathcal{I}(P) = \text{True}, \mathcal{I}(Q) = \text{False}$
- $\mathcal{I}(P) = \text{False}, \mathcal{I}(Q) = \text{True}$

Double-check with the truth tables!

P	Q	$P \vee Q$	$P \wedge Q$	$P \vee Q \supset P \wedge Q$	$\neg(P \vee Q \supset P \wedge Q)$
T	T	T	T	T	F
F	F	F	F	T	F
T	F	T	F	T	T
F	T	T	F	F	T

Homeworks!

Exercise

Show *unsatisfiability* of each of the following formulae using tableaux:

- $(p \equiv q) \equiv (\neg q \equiv p)$;
- $\neg((\neg q \supset \neg p) \supset ((\neg q \supset p) \supset q))$.

Show *satisfiability* of each of the following formulae using tableaux:

- $(p \equiv q) \supset (\neg q \equiv p)$;
- $\neg(p \vee q \supset ((\neg p \wedge q) \vee p \vee \neg q))$.

Show *validity* of each of the following formulae using tableaux:

- $(p \supset q) \supset ((p \supset \neg q) \supset \neg p)$;
- $(p \supset r) \supset (p \vee q \supset r \vee q)$.

For each of the following formulae, *describe all models* of this formula using tableaux:

- $(q \supset (p \wedge r)) \wedge \neg(p \vee r \supset q)$;
- $\neg((p \supset q) \wedge (p \wedge q \supset r) \supset (\neg p \supset r))$.

Establish the *equivalences* between the following pairs of formulae using tableaux:

- $(p \supset \neg p), \neg p$;
- $(p \supset q), (\neg q \supset \neg p)$;
- $(p \vee q) \wedge (p \vee \neg q), p$.

Termination

Assuming we analyse each formula at most once, we have:

Theorem (Termination)

For any propositional tableau, after a finite number of steps no more expansion rules will be applicable.

Hint for proof: This must be so, because each rule results in ever shorter formulas.

Note: Importantly, termination will *not* hold in the first-order case.

Soundness and Completeness

To actually believe that the tableau method is a valid decision procedure we have to prove:

Theorem (Soundness)

If $\Gamma \vdash \phi$ *then* $\Gamma \models \phi$

Theorem (Completeness)

If $\Gamma \models \phi$ *then* $\Gamma \vdash \phi$

Remember: We write $\Gamma \vdash \phi$ to say that there exists a closed tableau for $\Gamma \cup \{\neg\phi\}$.

Decidability

The proof of Soundness and Completeness confirms the decidability of propositional logic:

Theorem (Decidability)

The tableau method is a decision procedure for classical propositional logic.

Proof. To check validity of ϕ, develop a tableau for $\neg\phi$. Because of termination, we will eventually get a tableau that is either (1) closed or (2) that has a branch that cannot be closed.

- In case (1), the formula ϕ must be valid (soundness).
- In case (2), the branch that cannot be closed shows that $\neg\phi$ is satisfiable (see completeness proof), i.e. ϕ cannot be valid.

This terminates the proof.

Exercise

Exercise

Build a tableau for $\{(a \vee b) \wedge c, \neg b \vee \neg c, \neg a\}$