Artificial Intelligence & Machine Learning

A.Y. 2024/2025

Prof. Fabio Patrizi

Propositional Tableaux

How to Check Satisfiability

How to check whether there exists an interpretation I s.t. $I \models \varphi$?

We have seen Truth Tables: for complex formulas, approach is infeasible

We shall see two more approaches

Today: Tableaux method

Easy to understand, easy to implement

Tableaux

A Tableau is a tree-like structure, where:

- The root is the formula φ we want to check for satisfiability
- Internal nodes contain formulas over the propositions of φ
- Leaves are labelled with X or O

The tableau is constructed incrementally

Once construction is complete:

φ is satisfiable iff there exists some leaf labelled with O

equivalently, φ is unsatisfiable iff all leaves are labelled with X

Expansion rules

α-rules (or deterministic rules) PRODUCES ONE NODE

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

β-rules (or splitting rules) PRODUCES TWO NODES

$$\frac{\varphi \vee \psi}{\begin{array}{c|c}\varphi & \psi\end{array}} \qquad \begin{array}{c}\neg\varphi \vee \neg\psi \\ \dfrac{\neg(\varphi \wedge \psi)}{\begin{array}{c|c}\neg\varphi & \neg\psi\end{array}}\end{array} \qquad \begin{array}{c}\neg\varphi \vee \psi \\ \dfrac{\varphi \supset \psi}{\begin{array}{c|c}\neg\varphi & \psi\end{array}}\end{array} \qquad \begin{array}{c}\varphi \leftrightarrow \psi \equiv (\varphi \wedge \psi) \vee (\neg\varphi \wedge \neg\psi) \\ \dfrac{\varphi \equiv \psi}{\begin{array}{c|c}\varphi & \neg\varphi \\ \psi & \neg\psi\end{array}}\end{array} \qquad \frac{\neg(\varphi \equiv \psi)}{\begin{array}{c|c}\neg\varphi & \varphi \\ \psi & \neg\psi\end{array}}$$

$\neg(\varphi \leftrightarrow \psi) \equiv (\neg\varphi \wedge \psi) \vee (\varphi \wedge \neg\psi)$

Tableaux Construction Algorithm

Tableaux is constructed proceeding top-down

Start with root labelled with formula to check

At every step one node is expanded (nodes can be expanded only once)

Expansion order is arbitrary (but may affect size of Tableau)

If a path contains contradictory formulas, e.g., p and ¬p, add leaf with X

Once all nodes along a path have been expanded, paths with no contradiction terminate with O

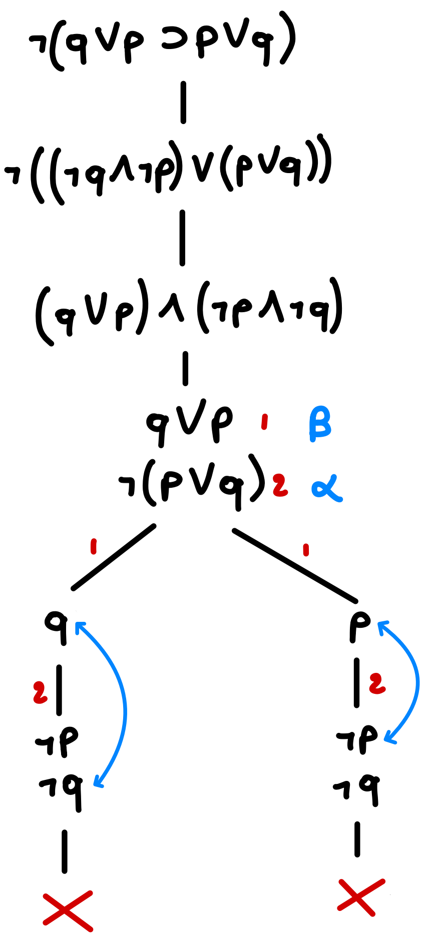
Construction is complete when every leaf has O or X

Example

All leaves contain X: unsatisfiable

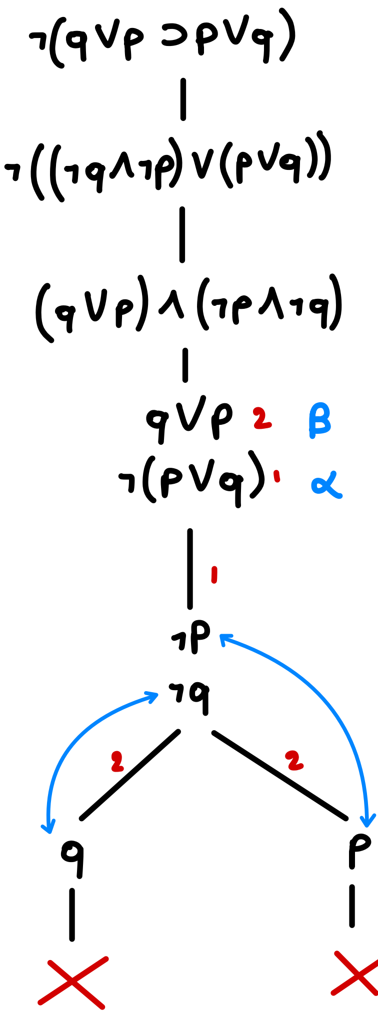
IT'S BETTER TO EXPAND α EARLIER THAN β 2)

1)



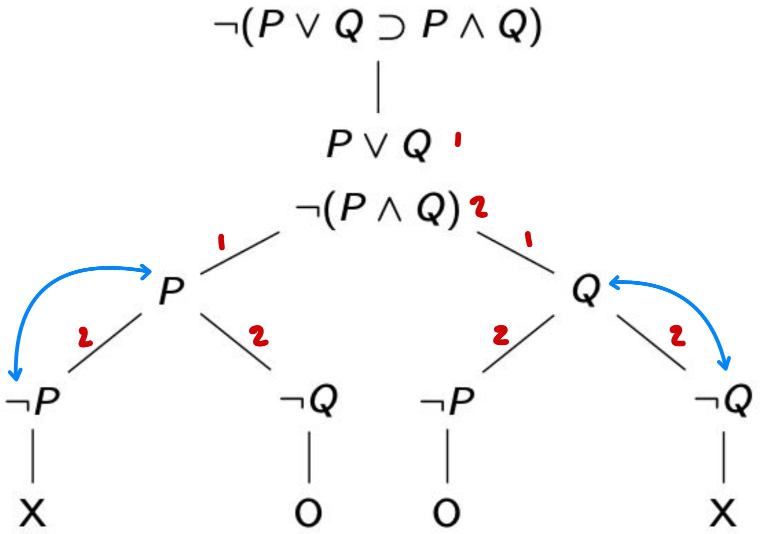
UNSAT

2)



Example

Some leaf contains O: satisfiable



SAT

Open and Closed Tableau

A root-leaf path, also called *branch*, is said to be:

- *closed* if it contains a formula and its negation
- *open* otherwise

A tableau is *closed* if all its branches are closed

Intuition

Every branch of a tableaux contains, overall, a set of formulas

If an interpretation I satisfies all the formulas in a branch, then $I \models \varphi$

By the construction of the tableau, also the converse holds: if $I \models \varphi$ then I satisfies all the formulas occurring in some branch of the tableau

Thus, $I \models \varphi$ iff I satisfies all the formulas in some branch of the tableau

Consequently, a closed tableau corresponds to an unsatisfiable φ

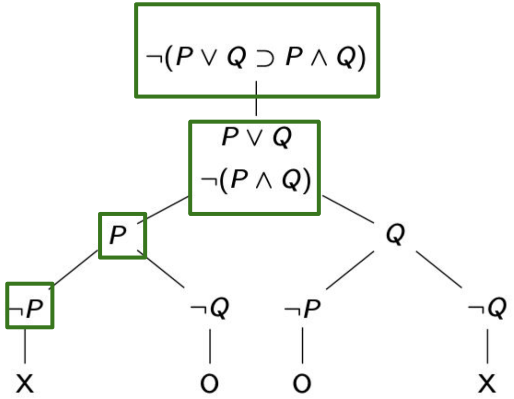
Models from Tableau

We can use literals to build models:

Traverse an open branch and collect all *direct literals*

(i.e., of the form p)

I={P} is a model of root formula



Satisfiability and Logical Implication

Recall the Refutation Principle: $\forall I$ SATISFIES $(I \models \Gamma \Rightarrow I \models \varphi)$

$$\Gamma \models \varphi \text{ iff } \Gamma \cup \{\neg\varphi\} \text{ is unsatisfiable}$$

If $\Gamma=\{\varphi_1,\ldots,\varphi_m\}$ is finite, this is equivalent to:

$$\Gamma \models \varphi \text{ iff } \varphi_1 \wedge \ldots \wedge \varphi_m \wedge \neg\varphi \text{ is unsatisfiable}$$

We can check whether $\Gamma \models \varphi$ by checking whether $\varphi_1 \wedge \ldots \wedge \varphi_m \wedge \neg\varphi$ is unsatisfiable

Can use any algorithm for satisfiability, including Tableaux

Special case $\Gamma=\varnothing$: $\models\varphi$ (φ is valid) iff $\neg\varphi$ is unsatisfiable