Mathematical Logics

FOL: Reasoning as deduction

Fausto Giunchiglia and Mattia Fumagalli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli

Lecture index

1. Reasoning problems (recap)
2. Hilbert systems (VAL – forward chaining)
3. Tableaux systems ((un)-SAT – backward chaining)
4. Correctness and completeness of Tableau
5. Examples
6. Termination
7. Countermodels

Tableaux Calculus

- The Tableaux Calculus is an algorithm solving the problem of satisfiability.
- If a formula is satisfiable, then there exists an open branch in the tableaux of this formula.
- The procedure attempts to construct the tableaux for a formula.
- ***Sometimes it is not possible to construct the tableaux since the model of the formula is infinite.***
- The basic idea is to incrementally build the model by looking at the formula, by decomposing it in a top/down fashion.
- The procedure exhaustively looks at all the possibilities, so that it can possibly prove that no model could be found for unsatisfiable formulas.

Same idea as in PL – extended to handle FOL Language

Propositonal tableaux (recap)

. . . for propositional connectives

α rules

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

β rules

$\varphi \vee \psi$		$\varphi \supset \psi$		$\neg(\varphi \wedge \psi)$		$\varphi \equiv \psi$		$\neg(\varphi \equiv \psi)$	
φ	ψ	$\neg\varphi$	ψ	$\neg\varphi$	$\neg\psi$	φ ψ	$\neg\varphi$ $\neg\psi$	φ $\neg\psi$	$\neg\varphi$ ψ

First order tableaux

Definition

A tableau is a rooted tree, where each node carries a first order sentence (closed formula), and the children of a node *n* are generated by applying a set of expansion rules to *n* or to one of the ancestors of *n*.

Definition

The expansion rules for a first order semantic tableaux are those for the propositional semantic tableaux, extended with the following rules that deal with the quantifiers:

γ rules

$$\frac{\forall x.\,\varphi(x)}{\varphi(t)} \qquad \frac{\neg\exists x.\,\varphi(x)}{\neg\varphi(t)}$$

Where t is a term free for x in φ

δ rules

$$\frac{\neg\forall x.\,\varphi(x)}{\neg\varphi(c)} \qquad \frac{\exists x.\,\varphi(x)}{\varphi(c)}$$

where c is a new constant not previously appearing in the tableaux

Substitution $\varphi[x/t]$

If $\varphi(x)$ is a free variable and t is a term, we use the notation $\varphi(t)$ instead of the more precise notation $\varphi[x/t]$ to represent the substitution of x for t in φ.

Substitution

$\varphi[x/t]$ denotes the formula we get by replacing each free occurrence of the variable x in the formula φ by the term t. This is admitted if t does not contain any variable y such that x occurs in the scope of a quantifier for y (i.e., in the scope of $\forall y$ or $\exists y$).

Substitution $\varphi[x/t]$

Example (of substitution)

$$P(x, y, f(x))[x/a] = P(a, y, f(a))$$

$$\forall x P(x, y)[x/b] = \forall x P(x, y)$$

$$\exists x P(x, x) \wedge Q(x)[x/c] = \exists x P(x, x) \wedge Q(c)$$

$$P(x, g(y))[y/f(x)] = P(x, g(f(x)))$$

$$\forall x.P(x, y)[y/f(x)] =$$ Not allowed since $f(x)$ is not free for y in $\forall x.P(x, y)$

Universal quantification rule

$$\frac{\forall x \,.\, \varphi(x)}{\varphi(t)}$$

- $\forall x \,.\, \varphi(x)$ means that for every object of the domain, the property $\varphi(x)$ should be true.
- a term t that occurs in the tableaux denotes an object of the domain
- therefore, $\varphi(t)$ must be true for all the terms t that occurs in the tableaux. I.e., the $\forall$ rule can be applied as many times as one want to any term that appears in the tableaux.

Exercize

Show that the following tableaux rule is sound.

$$\frac{\forall x \exists y P(y, x)}{\exists y P(y, f(x))}$$

Existential quantification rule

$$\frac{\exists x.\varphi(x)}{\varphi(c)} \quad \text{for a new constant } c$$

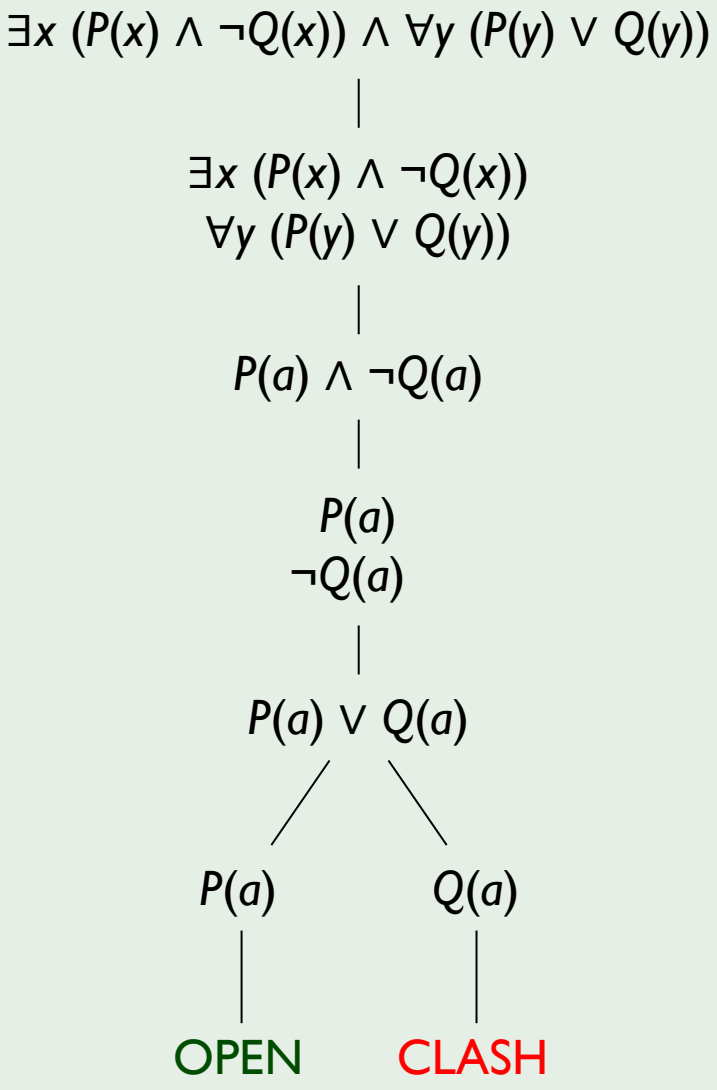
- $\exists x.\varphi(x)$: for some object of the domain $\varphi(x)$ should be true.
- But we don't know which object of the domain has the property φ, we know only that there is at least one.
- Therefore this rule cannot be applied to the terms that already occur in the tableaux, since otherwise we would introduce an unjustified property on the chosen element
- The trick is to introduce a term to denote an unconditioned object (sometimes called "fresh" constant/variable) for denoting an "unknown" object, i.e., an object on which we have no commitment.
- Therefore we allow only to infer $\varphi(c)$ form $\exists x.\varphi(x)$, where c is fresh.

Open and Closed Branches

- A tableaux rooted in φ is a method to search an interpretation that satisfies φ
- Every branch of a tableaux with root in φ, corresponds to an attempt to find an interpretation I that satisfies φ.
- The interpretation corresponding to a branch b of a tableaux should satisfy all the formulas that appear in the branch.
- If the branch contains two opposite literals, i.e. $P(t_1, \ldots, t_n)$ and $\neg P(t_1, \ldots, t_n)$, then the branch cannot correspond to an interpretation, since there is no interpretation that can satisfy one formula and its negated. So we can consider this attempt to find an interpretation failed. In this case we say that the branch is closed.
- If in a branch b all the rules have been applied and there is no opposite literals, then this branch corresponds to an interpretation. We call such a branch open

Example

Example



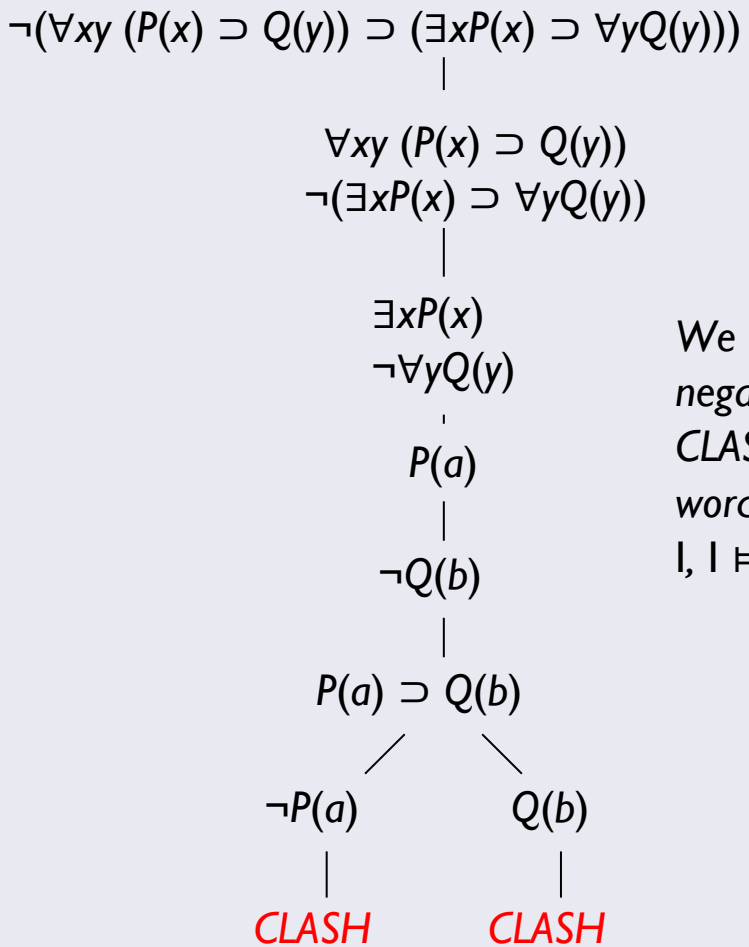
The tableaux method

- To test a set of formulas Γ is ***satisfiable***, start a tableau with all formulas in Γ. If the tableau closes off, then Γ is not satisfiable. If the tableau does not close off, then Γ is satisfiable, and from any open branch we can read off an interpretation satisfying Γ.
- A formula is ***unsatisfiable*** iff is not satisfiable
- To test a formula φ for ***validity***, start a tableau with $\neg\varphi$. If the tableau closes off, then φ is logically valid.
- To test whether φ is a ***logical consequence*** of Γ, start a tableau with all formulas in Γ and $\neg\varphi$. If the tableau closes off, then φ is indeed a logical consequence of Γ.
- Two formulas are ***logically equivalent*** if logical consequence holds in both directions

Example

Check via tableaux if the validity/satisfiability of the formula
$\varphi = \forall x, y\ (P(x) \supset Q(y)) \supset (\exists x P(x) \supset \forall y Q(y))$

Solution

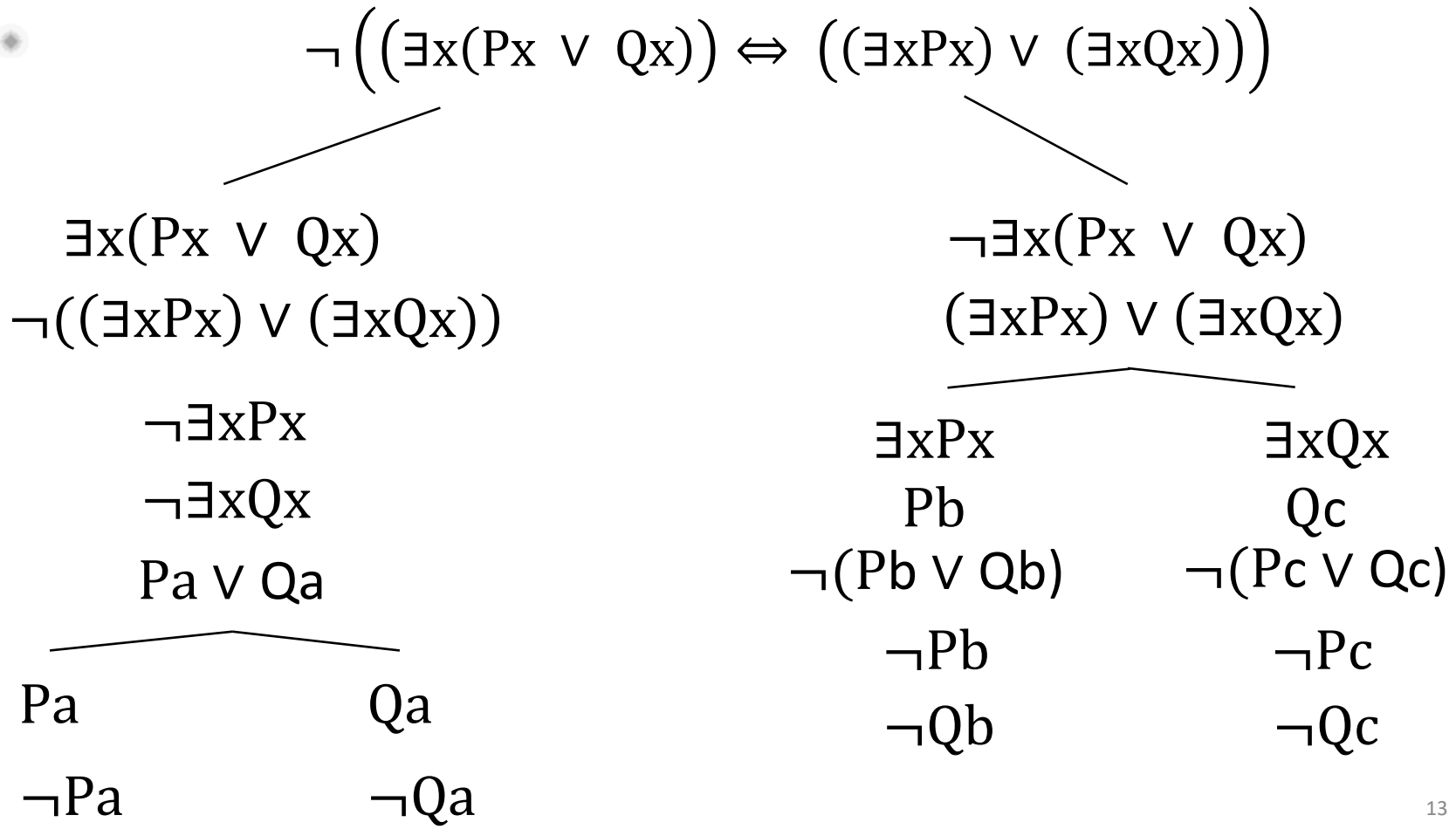


We try, vith the tableaux, to build a model for the negation of φ. Since the tableaux ends with all CLASHES, there is no such a model. In other words, for all I, I ⊭ ¬φ. Which implies that for all I, I ⊨ φ, i.e., that φ is valid.

Example

Example

To check if the formula $(\exists x(P(x) \vee Q(x))) \equiv ((\exists x P(x)) \vee (\exists x Q(x)))$ is satisfiable, we start with a tableaux with this formula:



Example (Cont'd)

Example

Check if $\forall x P(x) \wedge \exists x \neg P(f(x))$ is valid/satisfiable/unsatisfiable.

Solution

$\forall x P(x) \wedge \exists x \neg P(f(x))$

|

$\forall x P(x)$

$\exists x \neg P(f(x))$

|

$\neg P(f(c))$

|

$P(f(c))$

|

$\times$

Example

Example

Check if $\forall x P(x) \wedge \exists x \neg P(f(x))$ is valid/satisfiable/unsatisfiable.

Solution

$$\forall x P(x) \wedge \exists x \neg P(f(x))$$

$$\forall x P(x)$$
$$\exists x \neg P(f(x))$$

$$\neg P(f(c))$$

Now to expand $\forall x P(x)$, we can use any ground term t. Possible choices: c, $f(c)$, $f(f(c))$, ... we choose $f(c)$ because we want to create a clash with $\neg P(f(c))$. ***But possibility of a loop when searching!***

Mathematical Logics

FOL: Reasoning as deduction

Fausto Giunchiglia and Mattia Fumagalli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli