Reasoning in First Order Logic

Alessandro Artale

Free University of Bozen-Bolzano
Faculty of Computer Science
http://www.inf.unibz.it/˜artale

Descrete Mathematics and Logic — BSc course

Thanks to Prof. Enrico Franconi for provoding the slides

Reasoning Problems: Reduction to Satisfiability

Just like propositional logic:

- A formula ϕ is satisfiable iff there is some interpretation $\mathcal{I}$ that satisfies ϕ (i.e., ϕ is true under $\mathcal{I}$): $\mathcal{I} \models \phi$.
- Validity, equivalence, and entailment can be reduced to satisfiability:
  - ϕ is a valid (i.e., a tautology) iff $\neg\phi$ is not satisfiable.
  - ϕ is equivalent to ψ ($\phi \equiv \psi$) iff $\phi \leftrightarrow \psi$ is valid.
  - ϕ entails ψ ($\phi \models \psi$) iff $\phi \rightarrow \psi$ is valid (*deduction theorem*).
    - $\boxed{\phi \models \psi \text{ iff } \phi \land \neg\psi \text{ is not satisfiable}}$.
- We need a *sound and complete* procedure deciding satisfiability: the *tableaux calculus* is a decision procedure which checks the existence of a model.

Tableaux Calculus

Just like in propositional logic:

- The Tableaux Calculus is a decision procedure solving the problem of satisfiability.
- If a formula is satisfiable, the procedure will constructively exhibit a model of the formula.
- 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 eventually prove that no model could be found for unsatisfiable formulas.

Remark: With respect to propositional logic, the notion of *model* is different.

Tableaux Calculus: General Ideas

Finds a model for a given collection of sentences KB in **negation normal form**.

1. Consider the knowledge base KB as the root node of a *tree*.
2. Starting from the root, add new formulas to the tableaux, applying the *completion rules*.
3. Completion rules are either deterministic – they yield a uniquely determined successor node – or nondeterministic – yielding several possible alternative successor nodes (*branches*).
4. Apply the completion rules until either
   (a) an explicit contradiction due to the presence of two complementary ground literals in a node (a *clash*) is generated in each branch, or
   (b) there is a *completed* branch where no more rule is applicable (but. . .).

The completion rules for the propositional formulas:

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}}$$

If a model satisfies a conjunction, then it also satisfies *each of* the conjuncts

$$\frac{\phi \vee \psi}{\phi \mid \psi}$$

If a model satisfies a disjunction, then it also satisfies *one of* the disjuncts. It is a non-deterministic rule, and it generates two *alternative* branches of the tableaux.

The Calculus

The completion rules for quantified formulas:

$$\frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t]\\ \forall x.\ \phi\end{array}}$$

If a model satisfies a universal quantified formula, then it also satisfies the formula where the quantified variable has been substituted with all terms present in the Tableaux. Furthermore, the universal formula is not removed and appears also in the child node.

$$\frac{\exists x.\ \phi}{\phi[X/a]}$$

If a model satisfies an existential quantified formula, then it also satisfies the formula where the quantified variable has been substituted with a fresh new *skolem* constant.

Models

- The completed (open) branch of the Tableaux gives a model of *KB*: the *KB* is satisfiable.
  Since all formulas have been reduced to ground literals (i.e., either positive or negative atomic formulas which do not contain variables), it is possible to find an interpretation to predicates using the constants, which make all the sentences in the branch true.
- If there is no completed branch (i.e., every branch has a clash), then it is not possible to find an interpretation for the predicates making the original *KB* true: the *KB* is unsatisfiable.
  In fact, the original formulas from which the tree is constructed can not be true simultaneously.

Negation Normal Form

The above set of completion rules work only if the formula has been translated into Negation Normal Form, i.e.,

1. Eliminate $\rightarrow$ and $\leftrightarrow$, and
2. push down negations using the De Morgan rules:

$$\begin{aligned} \neg(\phi \vee \psi) &\equiv \neg\phi \wedge \neg\psi \\ \neg(\phi \wedge \psi) &\equiv \neg\phi \vee \neg\psi \\ \neg\forall x.\,\phi &\equiv \exists x.\,\neg\phi \\ \neg\exists x.\,\phi &\equiv \forall x.\,\neg\phi \end{aligned}$$

Example::
$\neg(\exists x.\ [\ \forall y.\ [\ P(x) \rightarrow Q(y)\ ]\ ])$
becomes
$\forall x.\ [\ \exists y.\ [\ P(x) \wedge \neg Q(y)\ ]\ ]$
(Why?)

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z.(p(z) \vee q(z))}$$

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi\\ \psi\end{array}} \quad \frac{\phi \vee \psi}{\phi \mid \psi} \quad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t]\\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (p(z) \vee q(z))}$$
$$\exists y.\ (p(y) \wedge \neg q(y))$$
$$\forall z.\ (p(z) \vee q(z))$$

Example

$$\frac{\phi \land \psi}{\begin{array}{c}\phi\\ \psi\end{array}} \qquad \frac{\phi \lor \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t]\\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \land \neg q(y)) \land \forall z. (p(z) \lor q(z))}$$

$$\exists y.\ (p(y) \land \neg q(y))$$

$$\forall z.\ (p(z) \lor q(z))$$

$$p(\mathbf{a}) \land \neg q(\mathbf{a})$$

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

Example

$$\frac{\phi \wedge \psi}{\begin{matrix}\phi \\ \psi\end{matrix}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{matrix}\phi[X/t] \\ \forall x.\ \phi\end{matrix}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (p(z) \vee q(z))}$$
$$\exists y.\ (p(y) \wedge \neg q(y))$$
$$\forall z.\ (p(z) \vee q(z))$$
$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$
$$p(\mathbf{a})$$
$$\neg q(\mathbf{a})$$
$$p(\mathbf{a}) \vee q(\mathbf{a})$$

Example

$$\frac{\phi \wedge \psi}{\begin{matrix}\phi \\ \psi\end{matrix}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{matrix}\phi[X/t] \\ \forall x.\ \phi\end{matrix}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

$$p(\mathbf{a}) \vee q(\mathbf{a})$$

$$p(\mathbf{a})$$

< COMPLETED >

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

$$p(\mathbf{a}) \vee q(\mathbf{a})$$

$p(\mathbf{a})$	$q(\mathbf{a})$
$< \texttt{COMPLETED} >$	$< \texttt{CLASH} >$

Example

$$\frac{\phi \land \psi}{\begin{array}{c}\phi\\ \psi\end{array}} \qquad \frac{\phi \lor \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t]\\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \land \neg q(y)) \land \forall z. (p(z) \lor q(z))}$$

$$\exists y.\ (p(y) \land \neg q(y))$$

$$\forall z.\ (p(z) \lor q(z))$$

$$p(\mathbf{a}) \land \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

$$p(\mathbf{a}) \lor q(\mathbf{a})$$

$p(\mathbf{a})$	$q(\mathbf{a})$
< COMPLETED >	< CLASH >

The formula is satisfiable. The devised model is $\mathbf{D} = \{\mathbf{a}\},\ p^{\mathcal{I}} = \{\mathbf{a}\},\ q^{\mathcal{I}} = \emptyset$.

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (\neg p(z) \vee q(z))}$$

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (\neg p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (\neg p(z) \vee q(z))$$

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (\neg p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (\neg p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z.(\neg p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (\neg p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z. (\neg p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (\neg p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

$$\neg p(\mathbf{a}) \vee q(\mathbf{a})$$

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi\\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t]\\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z.(\neg p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (\neg p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

$$\neg p(\mathbf{a}) \vee q(\mathbf{a})$$

$$\neg p(\mathbf{a})$$

$< \texttt{CLASH} >$

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z.(\neg p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (\neg p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

$$\neg p(\mathbf{a}) \vee q(\mathbf{a})$$

$\neg p(\mathbf{a})$	$q(\mathbf{a})$
< CLASH >	< CLASH >

Example

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\forall x.\ \phi}{\begin{array}{c}\phi[X/t] \\ \forall x.\ \phi\end{array}} \qquad \frac{\exists x.\ \phi}{\phi[X/a]}$$

$$\boxed{\exists y.\ (p(y) \wedge \neg q(y)) \wedge \forall z.(\neg p(z) \vee q(z))}$$

$$\exists y.\ (p(y) \wedge \neg q(y))$$

$$\forall z.\ (\neg p(z) \vee q(z))$$

$$p(\mathbf{a}) \wedge \neg q(\mathbf{a})$$

$$p(\mathbf{a})$$

$$\neg q(\mathbf{a})$$

$$\neg p(\mathbf{a}) \vee q(\mathbf{a})$$

$\neg p(\mathbf{a})$	$q(\mathbf{a})$
$<$ CLASH $>$	$<$ CLASH $>$

The formula is unsatisfiable.

Critical Points in a Tableaux Construction

- **Do not Use the Same Constant Twice to Instantiate Existential Formulas.**
  E.g., Try with the following Satisfiable formula:

  $$\forall x(p(x) \vee q(x)) \wedge \exists x. \neg p(x) \wedge \exists x. \neg q(x)$$

- **Instantiate Universal Formulas with all Constants.**
  Universal formulas cannot be disregarded once applied since they must be possibly ri-applied in case a new constant is introduced (e.g., see the Tableaux for the previous formula).
- **Infinite Branches.**
  E.g., Consider the following formula

  $$\forall x \exists y. p(x, y)$$

  Note. The method of semantic tableaux is not a decision procedure for satisfiability in first-order logic. Indeed, such a procedure does not exist in FOL, i.e., **FOL is undecidable**.

Infinite Models – Finite Model Property

There are FOL formulas that are satisfiable only in interpretations with an infinite Domain. In general, FOL is said not to have the Finite Model Property.

- The formula $\forall x \exists y. p(x, y)$ can be satisfied both in infinite models, e.g., $\mathcal{I}_{inf} = (\mathbf{Z}^+, \{<\})$, and in finite models, e.g., $\mathcal{I}_{fin} = (\{a\}, \{p(a, a)\})$.
- The following formula is satisfiable only over infinite models:

$$\forall x \exists y. p(x, y) \wedge \forall x. \neg p(x, x) \wedge \forall x \forall y \forall z (p(x, y) \wedge p(y, z) \rightarrow p(x, z))$$

Tableaux Construction

- **The Tableaux Construction Must be Systematic.**
  Consider the formula: $\forall x \exists y. p(x, y) \land \forall x(q(x) \land \neg q(x))$
  The formula is unsatisfiable but the Tableaux can decide to expand indefinitely the branch for the sub-formula $\forall x \exists y. p(x, y)$.
- A systematic construction is needed to make sure that rules are eventually applied to all formulas labeling a node.

Tableaux Construction (cont.)

- Initially, the Tableaux, $\mathcal{T}$, consists of the root node labeled with the formula, ϕ. If ϕ does not use constants we introduce an arbitrary constant, a_0.
- $\mathcal{T}$ is built inductively by repeatedly choosing an unmarked leaf, ℓ, and applying a Completion rule respecting the following order:
  1. If ℓ contains a complementary pair of literals (or $\perp$), mark the leaf closed (or clash);
  2. Apply an AND-rule if applicable;
  3. Apply an OR-rule if applicable;
  4. Apply an $\exists$-rule if applicable;
  5. Apply simultaneously all applicable $\forall$-rules with the proviso that only new formulas are added;
  6. If no rule applies mark the leaf open (or complete).

Tableaux Construction (cont.)

Definition. A branch in a Tableau is **closed** if it terminates in a leaf marked closed (clash); otherwise, if it is infinite or it terminates in a leaf marked open (complete), the branch is **open**.

A **Tableaux is closed** if all of its branches are closed; otherwise (it has a finite or infinite open branch), the **Tableaux is open**.

Termination and Decidability

Given a logic $\mathcal{L}$, a reasoning problem is said to be **decidable** if there exists a computational process (e.g., an algorithm, a computer program, etc.) that solves the problem in a **finite** number of steps, i.e., the process always terminates.

- The problem of deciding whether a formula φ is logically implied by a theory Γ is **undecidable** in FOL.
- Logical implication becomes decidable if we restrict to propositional calculus.
- Logical implication becomes decidable if we restrict to FOL using only at most *two* variable names; such language is called $\mathcal{L}_2$.

The property of (un)decidability is a general property of the problem and not of a particular algorithm solving it.

An example of $\mathcal{L}_3$ formula:
"The only baseball player in town is married to one of Jeremy's daughters."

$$
\begin{array}{l}
\exists \mathtt{x}.\ [\ \mathtt{B(x)} \wedge \\
\quad (\forall \mathtt{y}.\ [\ \mathtt{B(y)} \rightarrow \mathtt{x} = \mathtt{y}\ ]\ ) \wedge \\
\quad \exists \mathtt{z}.\ [\ \mathtt{M(x,z)} \wedge \mathtt{D(z,jeremy)} \wedge \\
\qquad (\exists \mathtt{k}.\ [\ \mathtt{D(k,jeremy)} \wedge \mathtt{z} \neq \mathtt{k}\ ]\ ) \wedge \\
\qquad \forall \mathtt{v}.\ [\ \mathtt{M(x,v)} \rightarrow \mathtt{v} = \mathtt{z}\ ]\ ]\ ]
\end{array}
$$

$$
\begin{array}{l}
\exists \mathtt{x}.\ [\ \mathtt{B(x)} \wedge \\
\quad (\forall \mathtt{y}.\ [\ \mathtt{B(y)} \rightarrow \mathtt{x} = \mathtt{y}\ ]\ ) \wedge \\
\quad \exists \mathtt{y}.\ [\ \mathtt{M(x,y)} \wedge \mathtt{D(y,jeremy)} \wedge \\
\qquad (\exists \mathtt{x}.\ [\ \mathtt{D(x,jeremy)} \wedge \mathtt{y} \neq \mathtt{x}\ ]\ ) \wedge \\
\qquad \forall \mathtt{v}.\ [\ \mathtt{M(x,v)} \rightarrow \mathtt{v} = \mathtt{y}\ ]\ ]\ ]
\end{array}
$$

Expressive Power

- Some logics can be made decidable by sacrificing some *expressive power*.
- A logical language $\mathcal{L}_a$ has more expressive power than a logical language $\mathcal{L}_b$, if each formula of $\mathcal{L}_b$ denotes the "same" set of models of its correspondent formula of $\mathcal{L}_a$, and if there is a formula of $\mathcal{L}_a$ denoting a set of models which is denoted by no formula in $\mathcal{L}_b$.

Example:
Consider $\mathcal{L}_a$ as FOL, and $\mathcal{L}_b$ as FOL without negation and disjunction. Given a common domain, the $\mathcal{L}_a$ formula $\exists x. \, [\ P(x) \vee Q(x)\ ]$ has a set of models which can not be captured by any formula of $\mathcal{L}_b$.
(Exercise: check it out with $\mathbf{D} = \{a\}$*)*

Tableaux Calculus: Correctness

Theorem—Soundness and Completeness. A formula ϕ in FOL is satisfiable if and only if the tableaux for ϕ contains an open (completed) branch.

Tableaux Calculus: Soundness

Soundness: If the Tableaux has one open (completed) branch then the formula is satisfiable.

To prove Soundness we need to show that the Interpretation that satisfies the set of literals labeling the nodes of a completed branch can be extended to a model of the formula labeling the root.

Tableaux Calculus: Soundness (cont.)

There are four steps in the proof (similar to the propositional case):

1. Define a property of formulas;
2. Show that the set of formulas in a completed branch has this property;
3. Prove that a set with this property is satisfiable;
4. Note that the formula in the root is in the set.

Tableaux Calculus: Soundness (cont.)

Step 1.

Definition–Hintikka Set. Let Θ be a set of closed formulas in FOL. Then Θ is a Hintikka set iff:

1. For all ground literals L appearing in a formula of Θ, either $L \notin \Theta$ or $\neg L \notin \Theta$.
2. If $\phi \wedge \psi \in \Theta$, then $\phi \in \Theta$ and $\psi \in \Theta$.
3. If $\phi \vee \psi \in \Theta$, then $\phi \in \Theta$ or $\psi \in \Theta$.
4. If $\forall x. \phi \in \Theta$, then $\phi[x/a] \in \Theta$ for *all* constants a in Θ.
5. If $\exists x. \phi \in \Theta$, then $\phi[x/a] \in \Theta$ for *some* constant a in Θ.

Tableaux Calculus: Soundness (cont.)

Step 2.

Lemma 1. If Θ is the union of formulas gathered on an open branch from the root to a completed leaf, then Θ is a Hintikka set.

Proof.

- If a grond literal appears at step n during the Tableaux construction, it will be preserved in any other node down to the leaf, Θ_l. Thus, all literals in Θ appear in Θ_l. Since the branch is completed, no complementary pair of literals appears in Θ_l, so condition (1) holds for Θ.
- For formulas different from $\forall x. \phi$, conditions 2, 3, or 5 easily hold since the branch is supposed to be completed, i.e., the completion rules have been applied.
- Consider a formula of the form $\forall x. \phi$. Since those formulas are never eliminated when the corresponding rule is applied, then the $\forall$-rule is applied to all possible constants in Θ and condition (4) holds.

Tableaux Calculus: Soundness (cont.)

Step 3.

Lamma 2. Let Θ be a Hintikka set. Then Θ is satisfiable.

- Let $\Delta = \{c_1, c_2, \ldots\}$ be the set of constants in Θ. We define an interpretation, $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, and then show that the interpretation is a model of Θ. For each n-ary predicate P:

$$
\begin{array}{ll}
(c_{i_1}, \ldots, c_{i_n}) \in P^{\mathcal{I}} & \text{if } P(c_{i_1}, \ldots, c_{i_n}) \in \Theta \\
(c_{i_1}, \ldots, c_{i_n}) \notin P^{\mathcal{I}} & \text{if } \neg P(c_{i_1}, \ldots, c_{i_n}) \in \Theta \\
(c_{i_1}, \ldots, c_{i_n}) \in P^{\mathcal{I}} & \text{otherwise}
\end{array}
$$

  By condition (1), $\mathcal{I}$ is well defined.

- By structural induction we can easily show that $\forall \phi \in \Theta,\ \mathcal{I} \models \phi$.

Tableaux Calculus: Soundness (cont.)

Step 4.

Proof of Soundness.

- Assume that ϕ has a Tableaux with a completed branch.
- By Lemma 1, Θ, the union of formulas on the nodes of that branch, is a Hintikka set.
- By Lemma 2, we can find an interprtation $\mathcal{I}$ for Θ.
- Since $\phi \in \Theta$, then $\mathcal{I} \models \phi$.

Tableaux Calculus: Completeness

Completeness: It is easier to prove the contrapositive: If the Tableaux has a clash in every branch, then the formula is unsatisfiable.

Proof. By induction on the height of the Tree generated by the Completion Rules.

- **Basis Case.** $h = 0$. Clearly a leaf is a set of literals and if it contains a clash it is unsatisfiable.
- **Induction Step.** By Inductive Hypothesis, for any node m, root of a tree with height $h_m < h_n$, the set of formulas, Θ_m, labeling node m is unsatisfiable if the Tree rooted at m, say T_m, contains a clash in every branch. The cases of AND-rule and OR-rule is as in propositional logic.

- **$\forall$-rule.** $\Theta_n = \{\forall x.\, \phi(x)\} \cup \Theta'_n$. Then, $\Theta_{n-1} = \{\forall x.\, \phi(x), \phi(a)\} \cup \Theta'_n$, with $h_{n-1} < h_n$. Now, if T_n has a clash in every branch so is T_{n-1} and **by induction** Θ_{n-1} is unsatisfiable.
  If, **by absurd**, Θ_n is satisfiable, then, there is a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that $\mathcal{I} \models \Theta_n$. Thus, $\mathcal{I} \models \Theta'_n$, and $\mathcal{I} \models \forall x.\, \phi(x)$. Since, by the Tableaux construction, the constant $a \in \Theta_n$, then $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and since $\mathcal{I} \models \forall x.\, \phi(x)$ then $\mathcal{I} \models \phi(a)$, which contradicts the fact that Θ_{n-1} is unsatisfiable.

- **∃-rule.** $\Theta_n = \{\exists x.\, \phi(x)\} \cup \Theta'_n$. Then, $\Theta_{n-1} = \{\phi(a)\} \cup \Theta'_n$, with $a \notin \Theta_n$ and $h_{n-1} < h_n$. Now, if T_n has a clash in every branch so is T_{n-1} and **by induction** Θ_{n-1} is unsatisfiable. **By absurd**, let Θ_n be satisfiable, i.e., there is a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that $\mathcal{I} \models \Theta_n$. Thus, $\mathcal{I} \models \Theta'_n$, and $\mathcal{I} \models \exists x.\, \phi(x)$. In particular, $\mathcal{I} \models \exists x.\, \phi(x)$, i.e., there exists $d \in \Delta^{\mathcal{I}}$ s.t. $\mathcal{I}, \alpha[x/d] \models \phi(x)$. Since $a \notin \Theta_n$, i.e., the constant a is fresh new, we can define a new interpretation, $\mathcal{I}'$, that extends $\mathcal{I}$ by mapping $a^{\mathcal{I}'} = d$. Then, $\mathcal{I}' \models \phi(a)$ and then $\mathcal{I}' \models \Theta_{n-1}$, contradicting the inductive hypothesis.

Other Reasoning Problems

- **Subsumption**
  - $\Gamma \models P \sqsubseteq Q$

    P and Q *predicate symbols* of the same arity
  - Q subsumes P in Γ iff $\Gamma \models \forall \hat{x}. \, [\, P(\hat{x}) \rightarrow Q(\hat{x}) \,]$

Other Reasoning Problems

- **Subsumption**
  - $\Gamma \models P \sqsubseteq Q$
    P and Q *predicate symbols* of the same arity
  - Q subsumes P in Γ iff $\Gamma \models \forall \hat{x}.\ [\ P(\hat{x}) \rightarrow Q(\hat{x})\ ]$
- **Instance Checking**
  - The constant a is an instance of the unary predicate P
  - $\Gamma \models P(a)$.

Example

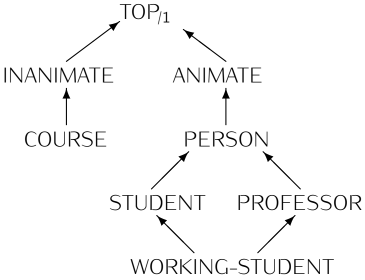
Given a theory Γ,
if $\texttt{STUDENT} \sqsubseteq \texttt{PERSON}$ is entailed by Γ
and if $\texttt{STUDENT(john)}$ is entailed by Γ
then $\texttt{PERSON(john)}$ is entailed by Γ

i.e., if

$\Gamma \models \texttt{STUDENT} \sqsubseteq \texttt{PERSON}$
$\Gamma \models \texttt{STUDENT(john)}$
then
$\Gamma \models \texttt{PERSON(john)}$

Subsumption

- *Subsumption* can be seen as a binary relation in the space of predicates with same arity: $P_{/n} \sqsubseteq Q_{/n}$.
- The subsumption relation is a *partial ordering* relation in the space of predicates of same arity.
  - **Exercise**: prove it.
  - *Hint*: a partial ordering relation is a transitive, reflexive, and antisymmetric relation.



Model Checking. Verify that a given Interpretation, $\mathcal{I}$, is a model for a closed formula φ: $\mathcal{I} \models \varphi$

Example:
$\Delta = \{a, b\}$,
$P^{\mathcal{I}} = \{a\}$,
$Q^{\mathcal{I}} = \{b\}$.
is a model of the formula:
$\exists y. \ [\ P(y) \wedge \neg Q(y)\ ] \wedge \forall z. [\ P(z) \vee Q(z)\ ]$

i.e.,
$\mathcal{I} \models \exists y. \ [\ P(y) \wedge \neg Q(y)\ ] \wedge \forall z. [\ P(z) \vee Q(z)\ ]$

Remark: Model Checking is decidable for FOL.

Summary

- Main lesson: Logic can be a useful tool. It allows us to represent information about a domain in a very straight-forward way then deduce additional facts using one general domain-independent "algorithm": deduction. Consequently, logic lends itself to large-scale, distributed-design problems.
- Each logic is made up of a syntax, a semantics, a definition of the reasoning problems and the computational properties, and inference procedures for the reasoning problems (possibly sound and complete). The syntax describes how to write correct sentences in the language, the semantics tells us what sentences mean in the "real world." The inference procedure derives results logically implied by a set of premises.

Summary (cont.)

- Logics differ in terms of their representation power and computational complexity of inference. The more restricted the representational power, the faster the inference in general.
- Propositional logic: we can only talk about facts and whether or not they are true. In the worst case, we can use the brute force truth-table method to do inference. Proof methods such as tableaux are generally more efficient, easier to implement, and easier to understand.
- First-order logic: we can now talk about objects and relations between them, and we can quantify over objects. Good for representing most interesting domains, but inference is not only expensive, but may not terminate.