Logic and Theoretical Foundation of Computer Science

LaTFoCS

Pamela Fleischmann

fpa@informatik.uni-kiel.de

Winter Semester 2019

Kiel University
Dependable Systems Group

Semantic Tableaux

Motivation

Truth tables are not an efficient decision procedure for satisfiability and by duality for validitiy!

- we need something better
- idea: decompose an arbitrary formula into smaller formulae

Literals and Complements

Definition

A literal is an atom or its negation. The atom is called positive literal and the negation is called negative literal. The set $\{p, \neg p\}$ for an atom p is called complementary pair of literals. Analogously $\{\varphi, \neg\varphi\}$ is called a complementary pair of the formula φ.

How do literals help for satisfiability?

Consider $\varphi = p \wedge (\neg q \vee \neg p)$.

- By definition of $\wedge$ we know $\beta(p) = \texttt{true}$.

How do literals help for satisfiability?

Consider $\varphi = p \wedge (\neg q \vee \neg p)$.

- By definition of $\wedge$ we know $\beta(p) = \texttt{true}$.
- By definition of $\vee$ we know $\beta(\neg q) = \texttt{true}$ or $\beta(\neg p) = \texttt{true}$.

How do literals help for satisfiability?

Consider $\varphi = p \wedge (\neg q \vee \neg p)$.

- By definition of $\wedge$ we know $\beta(p) = \texttt{true}$.
- By definition of $\vee$ we know $\beta(\neg q) = \texttt{true}$ or $\beta(\neg p) = \texttt{true}$.
- Combining both yields $\{p, \neg q\}$ needs to be satisfiable or $\{p, \neg p\}$ needs to be satisfiable

Satisfiability of Literal-Sets

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Satisfiability of Literal-Sets

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

The proof is postponed since we need to talk about contradiction proofs.

Contradiction Proofs

… or the power of dualism.

Contradiction Proofs

… or the power of dualism.

- in the world of classical logic we only have `true` and `false` and they are complementary

Contradiction Proofs

… or the power of dualism.

- in the world of classical logic we only have `true` and `false` and they are complementary
- each statement is either `true` or `false`

Contradiction Proofs

. . . or the power of dualism.

- in the world of classical logic we only have `true` and `false` and they are complementary
- each statement is either `true` or `false`
- if we want to prove that a statment is `true` we have basically two options

Contradiction Proofs

... or the power of dualism.

- in the world of classical logic we only have `true` and `false` and they are complementary
- each statement is either `true` or `false`
- if we want to prove that a statment is `true` we have basically two options
  1. starting with our axiom and deducing true smaller things step by step until we reach our claim

Contradiction Proofs

. . . or the power of dualism.

- in the world of classical logic we only have `true` and `false` and they are complementary
- each statement is either `true` or `false`
- if we want to prove that a statment is `true` we have basically two options
  1. starting with our axiom and deducing true smaller things step by step until we reach our claim
  2. Contradiction: suppose that the negation of the statement is true, deduce whatever you can until you find a contradiction (something excluded by the axioms or the definition etc.)

Assume and Suppose

- the words can be used interchangeably

Assume and Suppose

- the words can be used interchangeably
- in math we have a convention:

Assume and Suppose

- the words can be used interchangeably
- in math we have a convention:
  - **assume** if you don't want to get a contradiction

Assume and Suppose

- the words can be used interchangeably
- in math we have a convention:
  - **assume** if you don't want to get a contradiction
  - **suppose** if you want to get a contradiction

Assume and Suppose

- the words can be used interchangeably
- in math we have a convention:
  - **assume** if you don't want to get a contradiction
  - **suppose** if you want to get a contradiction
- advantage: the reader knows immediately in which direction the proof is going

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Rightarrow$.

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Rightarrow$.

- Ψ satisfiable set of literals

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Rightarrow$.

- Ψ satisfiable set of literals
- **Suppose:** there exists an atom p with $p, \neg p \in \Psi$

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Rightarrow$.

- Ψ satisfiable set of literals
- **Suppose:** there exists an atom p with $p, \neg p \in \Psi$
  - Ψ satisfiable $\rightsquigarrow$ there exists interpretation β with $\beta(q) = \mathtt{true}$ for all atoms $q \in \Psi$

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Rightarrow$.

- Ψ satisfiable set of literals
- **Suppose:** there exists an atom p with $p, \neg p \in \Psi$
  - Ψ satisfiable $\rightsquigarrow$ there exists interpretation β with $\beta(q) = \mathtt{true}$ for all atoms $q \in \Psi$
  - $p \in \Psi \rightsquigarrow \beta(p) = \mathtt{true}$

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Rightarrow$.

- Ψ satisfiable set of literals
- **Suppose:** there exists an atom p with $p, \neg p \in \Psi$
  - Ψ satisfiable $\rightsquigarrow$ there exists interpretation β with $\beta(q) = \mathtt{true}$ for all atoms $q \in \Psi$
  - $p \in \Psi \rightsquigarrow \beta(p) = \mathtt{true}$
  - $\neg p \in \Psi \rightsquigarrow \beta(\neg p) = \mathtt{true}$ and thus $\beta(p) = \mathtt{false}$

    Contradiction!

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Rightarrow$.

- Ψ satisfiable set of literals
- **Suppose:** there exists an atom p with $p, \neg p \in \Psi$
  - Ψ satisfiable $\rightsquigarrow$ there exists interpretation β with $\beta(q) = \texttt{true}$ for all atoms $q \in \Psi$
  - $p \in \Psi \rightsquigarrow \beta(p) = \texttt{true}$
  - $\neg p \in \Psi \rightsquigarrow \beta(\neg p) = \texttt{true}$ and thus $\beta(p) = \texttt{false}$ Contradiction!
- there does not exist an atom p with $p, \neg p \in \Psi$

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Leftarrow$.

- Ψ does not contain an atom p and its complementary

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Leftarrow$.

- Ψ does not contain an atom p and its complementary
- for each $q \in \Psi$ set $\beta(q) = \mathtt{true}$

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Leftarrow$.

- Ψ does not contain an atom p and its complementary
- for each $q \in \Psi$ set $\beta(q) = \texttt{true}$
- for each $\neg q \in \Psi$ set $\beta(q) = \texttt{false}$

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Leftarrow$.

- Ψ does not contain an atom p and its complementary
- for each $q \in \Psi$ set $\beta(q) = \texttt{true}$
- for each $\neg q \in \Psi$ set $\beta(q) = \texttt{false}$
- since Ψ is complementary-free, β is well-defined

Proof of the Theorem

Theorem

A set of literals is satisfiable iff it does not contain a complementary pair of literals.

Proof of $\Leftarrow$.

- Ψ does not contain an atom p and its complementary
- for each $q \in \Psi$ set $\beta(q) = \texttt{true}$
- for each $\neg q \in \Psi$ set $\beta(q) = \texttt{false}$
- since Ψ is complementary-free, β is well-defined
- Ψ is satisfiable $\square$

Negation Normal Form

Definition

A formula $\varphi \in \Phi$ is in negation normal form if $\neg$ occurs only directly in front of atoms.

Negation Normal Form

Definition

A formula $\varphi \in \Phi$ is in negation normal form if $\neg$ occurs only directly in front of atoms.

Example

$(\neg p \lor q) \land r \land \neg s$ is in NNF, while $\neg(p \lor q)$ is not

Negation Normal Form

Definition

A formula $\varphi \in \Phi$ is in negation normal form if $\neg$ occurs only directly in front of atoms.

Example

$(\neg p \vee q) \wedge r \wedge \neg s$ is in NNF, while $\neg(p \vee q)$ is not

Lemma

For each formula $\varphi \in \Phi$ exists an equivalent formula in NNF.

Algorithm for Constructing a Semantic Tableau

Input: formula $\varphi \in \Phi$ in NNF Let $\mathcal{T}$ be a tree with one unmarked node labeled with $\{\varphi\}$.
Repeat the following steps as long as possible:

1. Choose an unmarked leaf ℓ with the label $\Psi(\ell)$
2. Apply one of the following rules
   - 2.1 If $\Psi(\ell)$ only contains literals, mark it closed ($\times$) if it contains a complementary pair of literals and open ($\circ$) otherwise.
   - 2.2 If $\Psi(\ell)$ is not a set of literals, choose a formula $\psi \in \Psi$ that is not a literal and
     - 2.2.1 If $\psi = \chi_1 \wedge \chi_2$ then create a child ℓ' of ℓ and label it with $\Psi(\ell) \backslash \{\varphi\} \cup \{\chi_1, \chi_2\}$
     - 2.2.2 If $\psi = \chi_1 \vee \chi_2$ then create two children ℓ' and ℓ'' of ℓ labeled with $\Psi(\ell) \backslash \{\varphi\} \cup \{\chi_1\}$ and $\Psi(\ell) \backslash \{\varphi\} \cup \{\chi_2\}$ resp.

Open and Closed Tableaux

Definition

A tableau whose construction has terminated is called a completed tableau. A completed tableau is closed if all leaves are closed and open otherwise.

Some Thoughts about Algorithms

When we present an algorithm what do we have to do?

- Persuade the people that the algo works by threatening?

Some Thoughts about Algorithms

When we present an algorithm what do we have to do?

- Persuade the people that the algo works by threatening?
- Begging the people to believe that the algo works as intended?

Some Thoughts about Algorithms

When we present an algorithm what do we have to do?

- Persuade the people that the algo works by threatening?
- Begging the people to believe that the algo works as intended?
- NO! we have to prove a couple of things!

Some Thoughts about Algorithms

When we present an algorithm what do we have to do?

- Persuade the people that the algo works by threatening?
- Begging the people to believe that the algo works as intended?
- NO! we have to prove a couple of things!
  - Termination

Some Thoughts about Algorithms

When we present an algorithm what do we have to do?

- Persuade the people that the algo works by threatening?
- Begging the people to believe that the algo works as intended?
- NO! we have to prove a couple of things!
  - Termination
  - Completeness: if the input is correct, the algo returns a correct output

Some Thoughts about Algorithms

When we present an algorithm what do we have to do?

- Persuade the people that the algo works by threatening?
- Begging the people to believe that the algo works as intended?
- NO! we have to prove a couple of things!
  - Termination
  - Completeness: if the input is correct, the algo returns a correct output
  - Soundness: if the algo returns a correct output, the input have been correct

Some Thoughts about Algorithms

When we present an algorithm what do we have to do?

- Persuade the people that the algo works by threatening?
- Begging the people to believe that the algo works as intended?
- NO! we have to prove a couple of things!
  - Termination
  - Completeness: if the input is correct, the algo returns a correct output
  - Soundness: if the algo returns a correct output, the input have been correct
  - Complexity: time and space

Correctness means sound and complete.

Termination

Sketch of Proof:

- in step 2.1 sets of literals are marked

Termination

Sketch of Proof:

- in step 2.1 sets of literals are marked
- the algorithm terminates if all leaves are marked

Termination

Sketch of Proof:

- in step 2.1 sets of literals are marked
- the algorithm terminates if all leaves are marked
- In step 2.2 leaves with strictly shorter formulae are attached

Termination

Sketch of Proof:

- in step 2.1 sets of literals are marked
- the algorithm terminates if all leaves are marked
- In step 2.2 leaves with strictly shorter formulae are attached
- getting strictly shorter implies that at some point the literals are reached

Termination

Sketch of Proof:

- in step 2.1 sets of literals are marked
- the algorithm terminates if all leaves are marked
- In step 2.2 leaves with strictly shorter formulae are attached
- getting strictly shorter implies that at some point the literals are reached
- $\rightsquigarrow$ termination $\square$

Soundness

We have to prove

Lemma

If $\mathcal{T}_\varphi$ is closed then φ is unsatisfiable.

Soundness

We have to prove

Lemma

If $\mathcal{T}_\varphi$ is closed then φ is unsatisfiable.

we are going to prove something more general:

Lemma

If the subtree $\mathcal{T}_n$ rooted at node n is closed, then the label $U(n)$ at n is unsatisfiable.

Proof of Soundness

- Induction on height h_n of node n

Proof of Soundness

- Induction on height h_n of node n
- $h_n = 0$ implies that n is a leaf

Proof of Soundness

- Induction on height h_n of node n
- $h_n = 0$ implies that n is a leaf
- leaf closed $\rightsquigarrow$ label is unsatisfiable

Proof of Soundness

- Induction on height h_n of node n
- $h_n = 0$ implies that n is a leaf
- leaf closed $\rightsquigarrow$ label is unsatisfiable
- Assume that for all $m < n$, the label of $\mathcal{T}_m$ is unsatisfiable if $\mathcal{T}_m$ is closed, for a fixed but arbitrary $n \in \mathbb{N}$.

Proof of Soundness

- Induction on height h_n of node n
- $h_n = 0$ implies that n is a leaf
- leaf closed $\rightsquigarrow$ label is unsatisfiable
- Assume that for all $m < n$, the label of $\mathcal{T}_m$ is unsatisfiable if $\mathcal{T}_m$ is closed, for a fixed but arbitrary $n \in \mathbb{N}$.
- n not a leaf $\rightsquigarrow$ n has one child or two children

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0
- the label n' of the child contains $\Psi_0 \cup \{\varphi_1, \varphi_2\}$

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0
- the label n' of the child contains $\Psi_0 \cup \{\varphi_1, \varphi_2\}$
- $\mathcal{T}_n$ closed $\rightsquigarrow \mathcal{T}_{n'}$ closed $\rightsquigarrow \Psi_0 \cup \{\varphi_1, \varphi_2\}$ unsatisfiable by (IH)

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0
- the label n' of the child contains $\Psi_0 \cup \{\varphi_1, \varphi_2\}$
- $\mathcal{T}_n$ closed $\rightsquigarrow$ $\mathcal{T}_{n'}$ closed $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1, \varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0
- the label n' of the child contains $\Psi_0 \cup \{\varphi_1, \varphi_2\}$
- $\mathcal{T}_n$ closed $\rightsquigarrow$ $\mathcal{T}_{n'}$ closed $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1, \varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
    - $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1 \wedge \varphi_2)$ unsatisfiable

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0
- the label n' of the child contains $\Psi_0 \cup \{\varphi_1, \varphi_2\}$
- $\mathcal{T}_n$ closed $\rightsquigarrow$ $\mathcal{T}_{n'}$ closed $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1, \varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
  - $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1 \wedge \varphi_2)$ unsatisfiable
- case 2: Ψ_0 satisfiable

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0
- the label n' of the child contains $\Psi_0 \cup \{\varphi_1, \varphi_2\}$
- $\mathcal{T}_n$ closed $\leadsto$ $\mathcal{T}_{n'}$ closed $\leadsto$ $\Psi_0 \cup \{\varphi_1, \varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
  - $\leadsto$ $\Psi_0 \cup \{\varphi_1 \wedge \varphi_2)$ unsatisfiable
- case 2: Ψ_0 satisfiable
  - $\leadsto$ $\{\varphi_1, \varphi_2\}$ unsatisfiable

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0
- the label n' of the child contains $\Psi_0 \cup \{\varphi_1, \varphi_2\}$
- $\mathcal{T}_n$ closed $\rightsquigarrow$ $\mathcal{T}_{n'}$ closed $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1, \varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
  - $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1 \wedge \varphi_2)$ unsatisfiable
- case 2: Ψ_0 satisfiable
  - $\rightsquigarrow$ $\{\varphi_1, \varphi_2\}$ unsatisfiable
  - $\rightsquigarrow$ w.l.o.g. φ_1 unsatisfiable

Proof of Soundness: 1 Child

- the label of n contains $\varphi_1 \wedge \varphi_2$ and a set of formulae Ψ_0
- the label n' of the child contains $\Psi_0 \cup \{\varphi_1, \varphi_2\}$
- $\mathcal{T}_n$ closed $\rightsquigarrow$ $\mathcal{T}_{n'}$ closed $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1, \varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
  - $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1 \wedge \varphi_2)$ unsatisfiable
- case 2: Ψ_0 satisfiable
  - $\rightsquigarrow$ $\{\varphi_1, \varphi_2\}$ unsatisfiable
  - $\rightsquigarrow$ w.l.o.g. φ_1 unsatisfiable
  - $\rightsquigarrow$ $\varphi_1 \wedge \varphi_2$ unsatisfiable $\rightsquigarrow$ $\Psi_0 \cup \{\varphi_1 \wedge \varphi_2\}$ unsatisfiable

Proof of Soundness: 2 Children

- the label contains $\varphi_1 \vee \varphi_2$ and a set of formulae Ψ_0

Proof of Soundness: 2 Children

- the label contains $\varphi_1 \vee \varphi_2$ and a set of formulae Ψ_0
- the labels of the children n_1, n_2 contain $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ resp.

Proof of Soundness: 2 Children

- the label contains $\varphi_1 \vee \varphi_2$ and a set of formulae Ψ_0
- the labels of the children n_1, n_2 contain $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ resp.
- $\mathcal{T}_n$ closed $\rightsquigarrow \mathcal{T}_{n_1}, \mathcal{T}_{n_2}$ closed $\rightsquigarrow \Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ unsatisfiable by (IH)

Proof of Soundness: 2 Children

- the label contains $\varphi_1 \vee \varphi_2$ and a set of formulae Ψ_0
- the labels of the children n_1, n_2 contain $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ resp.
- $\mathcal{T}_n$ closed $\rightsquigarrow \mathcal{T}_{n_1}, \mathcal{T}_{n_2}$ closed $\rightsquigarrow \Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable

Proof of Soundness: 2 Children

- the label contains $\varphi_1 \vee \varphi_2$ and a set of formulae Ψ_0
- the labels of the children n_1, n_2 contain $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ resp.
- $\mathcal{T}_n$ closed $\rightsquigarrow \mathcal{T}_{n_1}, \mathcal{T}_{n_2}$ closed $\rightsquigarrow \Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
  - $\rightsquigarrow \Psi_0 \cup \{\varphi_1 \wedge \varphi_2)$ unsatisfiable

Proof of Soundness: 2 Children

- the label contains $\varphi_1 \lor \varphi_2$ and a set of formulae Ψ_0
- the labels of the children n_1, n_2 contain $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ resp.
- $\mathcal{T}_n$ closed $\leadsto$ $\mathcal{T}_{n_1}, \mathcal{T}_{n_2}$ closed $\leadsto$ $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
  - $\leadsto$ $\Psi_0 \cup \{\varphi_1 \land \varphi_2)$ unsatisfiable
- case 2: Ψ_0 satisfiable

Proof of Soundness: 2 Children

- the label contains $\varphi_1 \vee \varphi_2$ and a set of formulae Ψ_0
- the labels of the children n_1, n_2 contain $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ resp.
- $\mathcal{T}_n$ closed $\leadsto$ $\mathcal{T}_{n_1}, \mathcal{T}_{n_2}$ closed $\leadsto$ $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
  - $\leadsto$ $\Psi_0 \cup \{\varphi_1 \wedge \varphi_2)$ unsatisfiable
- case 2: Ψ_0 satisfiable
  - $\leadsto$ $\{\varphi_1\}$ and $\{\varphi_2\}$ unsatisfiable

Proof of Soundness: 2 Children

- the label contains $\varphi_1 \vee \varphi_2$ and a set of formulae Ψ_0
- the labels of the children n_1, n_2 contain $\Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ resp.
- $\mathcal{T}_n$ closed $\rightsquigarrow \mathcal{T}_{n_1}, \mathcal{T}_{n_2}$ closed $\rightsquigarrow \Psi_0 \cup \{\varphi_1\}$ and $\Psi_0 \cup \{\varphi_2\}$ unsatisfiable by (IH)
- case 1: Ψ_0 unsatisfiable
  - $\rightsquigarrow \Psi_0 \cup \{\varphi_1 \wedge \varphi_2)$ unsatisfiable
- case 2: Ψ_0 satisfiable
  - $\rightsquigarrow \{\varphi_1\}$ and $\{\varphi_2\}$ unsatisfiable
  - $\rightsquigarrow \varphi_1 \vee \varphi_2$ unsatisfiable $\rightsquigarrow \Psi_0 \cup \{\varphi_1 \vee \varphi_2\}$ unsatisfiable $\square$

Completeness

We have to prove

Lemma

If φ is unsatisfiable then all semantic trees are closed.

Completeness

We have to prove

Lemma

If φ is unsatisfiable then all semantic trees are closed.

Before we can prove this lemma, we have to talk about contraposition.

Contraposition

- Contraposition is a proving technique for implications.

Contraposition

- Contraposition is a proving technique for implications.
- Consider $A \to B$.

Contraposition

- Contraposition is a proving technique for implications.
- Consider $A \rightarrow B$.
- We know by truth table: $A \rightarrow B \equiv \neg B \rightarrow \neg A$

Contraposition

- Contraposition is a proving technique for implications.
- Consider $A \longrightarrow B$.
- We know by truth table: $A \longrightarrow B \equiv \neg B \longrightarrow \neg A$
- thus we can also prove $\neg B \longrightarrow \neg A$ instead of proving $A \longrightarrow B$

Contraposition

- Contraposition is a proving technique for implications.
- Consider $A \longrightarrow B$.
- We know by truth table: $A \longrightarrow B \equiv \neg B \longrightarrow \neg A$
- thus we can also prove $\neg B \longrightarrow \neg A$ instead of proving $A \longrightarrow B$
- this is often easier especially if we have to deal with general statements

Proof of Completeness

We are going to prove

Lemma

If one not closed semantic tree exists, φ is satisfiable.

Proof of Completeness

We are going to prove

Lemma

If one not closed semantic tree exists, φ is satisfiable.

Plan for the Proof:

Proof of Completeness

We are going to prove

Lemma

If one not closed semantic tree exists, φ is satisfiable.

Plan for the Proof:

- if the semantic tree is not closed $\rightsquigarrow$ there is an open leaf

Proof of Completeness

We are going to prove

Lemma

If one not closed semantic tree exists, φ is satisfiable.

Plan for the Proof:

- if the semantic tree is not closed $\leadsto$ there is an open leaf
- we are extending the interpretation for the leaf to an interpretation of φ

Proof of Completeness

We are going to prove

Lemma

If one not closed semantic tree exists, φ is satisfiable.

Plan for the Proof:

- if the semantic tree is not closed $\rightsquigarrow$ there is an open leaf
- we are extending the interpretation for the leaf to an interpretation of φ
- we do this by induction on the length of the branch

Proof of Completeness (Cont)

We have four steps in the proof:

1. Define a property of sets of formulae.

Proof of Completeness (Cont)

We have four steps in the proof:

1. Define a property of sets of formulae.
2. Show that the union of the formula labeling nodes in an open branch has this property.

Proof of Completeness (Cont)

We have four steps in the proof:

1. Define a property of sets of formulae.
2. Show that the union of the formula labeling nodes in an open branch has this property.
3. Prove that any set having this property is satisfiable.

Proof of Completeness (Cont)

We have four steps in the proof:

1. Define a property of sets of formulae.
2. Show that the union of the formula labeling nodes in an open branch has this property.
3. Prove that any set having this property is satisfiable.
4. Note that the formula labeling the root is in the set.

Proof of Completeness (Cont)

Definition

A set $\Psi \subseteq \Phi$ is a Hintikka set iff

1. for each atom p in a formula in Ψ, either $p \notin \Psi$ or $\neg p \notin \Psi$
2. If $\varphi_1 \wedge \varphi_2 \in \Psi$ implies $\varphi_1, \varphi_2 \in \Psi$ and if $\neg\neg\varphi_1 \in \Psi$ then $\varphi_1 \in \Psi$
3. If $\varphi_1 \vee \varphi_2 \in \Psi$ implies $\varphi_1 \in \Psi$ or $\varphi_2 \in \Psi$.

Proof of Completeness (Cont)

Lemma

Let ℓ be an open leaf in a tableau for φ. Set $\Psi = \bigcup_i \Psi(i)$ where i runs over the set of nodes on the branch from the root to ℓ. Then Ψ is a Hintikka set.

Proof of Completeness (Cont)

Lemma

Let ℓ be an open leaf in a tableau for φ. Set $\Psi = \bigcup_i \Psi(i)$ where i runs over the set of nodes on the branch from the root to ℓ. Then Ψ is a Hintikka set.

Proof.

Proof of Completeness (Cont)

Lemma

Let ℓ be an open leaf in a tableau for φ. Set $\Psi = \bigcup_i \Psi(i)$ where i runs over the set of nodes on the branch from the root to ℓ. Then Ψ is a Hintikka set.

Proof.

- if one $\Psi(i)$ contains a literal, all labels of the children contain this literal

Proof of Completeness (Cont)

Lemma

Let ℓ be an open leaf in a tableau for φ. Set $\Psi = \bigcup_i \Psi(i)$ where i runs over the set of nodes on the branch from the root to ℓ. Then Ψ is a Hintikka set.

Proof.

- if one $\Psi(i)$ contains a literal, all labels of the children contain this literal
- ℓ open $\rightsquigarrow$ no $\Psi(i)$ contains a complementary pair (condition 1)

Proof of Completeness (Cont)

Lemma

Let ℓ be an open leaf in a tableau for φ. Set $\Psi = \bigcup_i \Psi(i)$ where i runs over the set of nodes on the branch from the root to ℓ. Then Ψ is a Hintikka set.

Proof.

- if one $\Psi(i)$ contains a literal, all labels of the children contain this literal
- ℓ open $\leadsto$ no $\Psi(i)$ contains a complementary pair (condition 1)
- by the definition of the semantic tableau condition 2. and 3. hold $\square$

Proof of Completeness (Cont)

Lemma (Hintikka)

Every Hintikka set is satisfiable.

Proof of Completeness (Cont)

Lemma (Hintikka)

Every Hintikka set is satisfiable.

Proof.

Proof of Completeness (Cont)

Lemma (Hintikka)

Every Hintikka set is satisfiable.

Proof.

- Ψ Hintikka set, A_Ψ set of all atoms occuring in a formula of Ψ

Proof of Completeness (Cont)

Lemma (Hintikka)

Every Hintikka set is satisfiable.

Proof.

- Ψ Hintikka set, A_Ψ set of all atoms occuring in a formula of Ψ
- define β by $\beta(p) = \begin{cases} \texttt{true} & \text{if } p \in \Psi \vee (p \notin \Psi \wedge \neg p \notin \Psi) \\ \texttt{false} & \text{if } \neg p \in \Psi. \end{cases}$

Proof of Completeness (Cont)

Lemma (Hintikka)

Every Hintikka set is satisfiable.

Proof.

- Ψ Hintikka set, A_Ψ set of all atoms occuring in a formula of Ψ
- define β by $\beta(p) = \begin{cases} \texttt{true} & \text{if } p \in \Psi \vee (p \notin \Psi \wedge \neg p \notin \Psi) \\ \texttt{false} & \text{if } \neg p \in \Psi. \end{cases}$
- we have to prove by structural induction $\hat{\beta}(\varphi) = \texttt{true}$

Proof of Completeness (Cont)

- φ atom $\rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$

Proof of Completeness (Cont)

- φ atom $\rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$
- $\varphi = \neg p \rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$

Proof of Completeness (Cont)

- φ atom $\rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$
- $\varphi = \neg p \rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$
- $\varphi = \varphi_1 \land \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \texttt{true}$

Proof of Completeness (Cont)

- φ atom $\rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \neg p \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \varphi_1 \wedge \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{true}$
- by condition 2. $\varphi_1, \varphi_2 \in \Psi$ and by IH $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{true}$ $\rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$

Proof of Completeness (Cont)

- φ atom $\rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \neg p \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \varphi_1 \wedge \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{true}$
- by condition 2. $\varphi_1, \varphi_2 \in \Psi$ and by IH $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{true}$ $\rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \varphi_1 \vee \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{false}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{false}$

Proof of Completeness (Cont)

- φ atom $\rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \neg p \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \varphi_1 \wedge \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{true}$
- by condition 2. $\varphi_1, \varphi_2 \in \Psi$ and by IH $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{true}$ $\rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \varphi_1 \vee \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{false}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{false}$
- by condition 2. φ_1 or φ_2 are elements of Ψ

Proof of Completeness (Cont)

- φ atom $\rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \neg p \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \varphi_1 \wedge \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{true}$
- by condition 2. $\varphi_1, \varphi_2 \in \Psi$ and by IH $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{true}$ $\rightsquigarrow \hat{\beta}(\varphi) = \mathtt{true}$
- $\varphi = \varphi_1 \vee \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \mathtt{false}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \mathtt{false}$
- by condition 2. φ_1 or φ_2 are elements of Ψ
- by IH $\hat{\beta}(\varphi_1) = \mathtt{true}$ or $\hat{\beta}(\varphi_2) = \mathtt{true}$

Proof of Completeness (Cont)

- φ atom $\rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$
- $\varphi = \neg p \rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$
- $\varphi = \varphi_1 \wedge \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \texttt{true}$
- by condition 2. $\varphi_1, \varphi_2 \in \Psi$ and by IH $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \texttt{true}$ $\rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$
- $\varphi = \varphi_1 \vee \varphi_2 \rightsquigarrow \hat{\beta}(\varphi) = \texttt{false}$ iff $\hat{\beta}(\varphi_1) = \hat{\beta}(\varphi_2) = \texttt{false}$
- by condition 2. φ_1 or φ_2 are elements of Ψ
- by IH $\hat{\beta}(\varphi_1) = \texttt{true}$ or $\hat{\beta}(\varphi_2) = \texttt{true}$
- $\rightsquigarrow \hat{\beta}(\varphi) = \texttt{true}$ $\square$

Proof of Completeness (Cont)

Proof of Completeness.

- $\mathcal{T}$ open tableau for φ

Proof of Completeness (Cont)

Proof of Completeness.

- $\mathcal{T}$ open tableau for φ
- union of the labels from open node to branch is Hintikka set

Proof of Completeness (Cont)

Proof of Completeness.

- $\mathcal{T}$ open tableau for φ
- union of the labels from open node to branch is Hintikka set
- the union of the labels is the label of the roots, thus φ

Proof of Completeness.

- $\mathcal{T}$ open tableau for φ
- union of the labels from open node to branch is Hintikka set
- the union of the labels is the label of the roots, thus φ
- φ satisfiable $\square$

Consequences from Correctness

Corollary

φ is satisfiable iff $\mathcal{T}$ is open.

Consequences from Correctness

Corollary

φ is satisfiable iff $\mathcal{T}$ is open.

Corollary

φ is valid iff $\mathcal{T}_{\neg\varphi}$ is closed.

Consequences from Correctness

Corollary

φ is satisfiable iff $\mathcal{T}$ is open.

Corollary

φ is valid iff $\mathcal{T}_{\neg\varphi}$ is closed.

Corollary

Semantic Tableaux is a decision procedure for validity in propositional logic.