Lecture 2: Analysis of the Tableau Algorithm

Dr. Son P. Nguyen

UEL
VNU-HCMC

January 20, 2016

Table of Contents

Definitions and Proofs by Structural Induction

Definitions by Structural Induction

Goal: Define a function f on the set of all propositional formulae

By induction on the structure of propositional formulae:

1. Base Case:
   Define $f(p_i)$ for all proposition symbols p_i.
2. Inductive Step:
   For all propositional formulae φ and ψ, define

$$f(\neg\varphi), \qquad f(\varphi \wedge \psi), \qquad f(\varphi \vee \psi)$$

in terms of $f(\varphi)$ and $f(\psi)$.

Subformulae

Definition of the set $\mathsf{sub}(\varphi)$ of all subformulae of φ:

1. Base Case: For all integers $i \geq 1$, define:

$$\mathsf{sub}(p_i) := \{p_i\}$$

2. Inductive Step: For all propositional formulae φ and ψ, let:

$$\begin{aligned}
\mathsf{sub}(\neg\varphi) &:= \{\neg\varphi\} \cup \mathsf{sub}(\varphi) \\
\mathsf{sub}(\varphi \wedge \psi) &:= \{\varphi \wedge \psi\} \cup \mathsf{sub}(\varphi) \cup \mathsf{sub}(\psi) \\
\mathsf{sub}(\varphi \vee \psi) &:= \{\varphi \vee \psi\} \cup \mathsf{sub}(\varphi) \cup \mathsf{sub}(\psi)
\end{aligned}$$

Subformulae example

For $\varphi = (p_1 \wedge \neg p_2) \vee \neg p_3$, we have:

$$
\begin{aligned}
\mathsf{sub}(\varphi) &= \{\varphi\} \cup \mathsf{sub}(p_1 \wedge \neg p_2) \cup \mathsf{sub}(\neg p_3) \\
&= \{\varphi\} \cup \{p_1 \wedge \neg p_2\} \cup \mathsf{sub}(p_1) \cup \mathsf{sub}(\neg p_2) \cup \mathsf{sub}(\neg p_3) \\
&= \{\varphi, p_1 \wedge \neg p_2\} \cup \{p_1\} \cup \{\neg p_2\} \cup \mathsf{sub}(p_2) \cup \{\neg p_3\} \cup \mathsf{sub}(p_3) \\
&= \{\varphi, p_1 \wedge \neg p_2\} \cup \{p_1\} \cup \{\neg p_2\} \cup \{p_2\} \cup \{\neg p_3\} \cup \{p_3\} \\
&= \{\varphi, p_1 \wedge \neg p_2, p_1, \neg p_2, p_2, \neg p_3, p_3\}.
\end{aligned}
$$

Length of formulae

Definition of the length $\mathrm{len}(\varphi)$ of a formula φ:

- $\mathrm{len}(p_i) := 1$ for all integers $i \geq 1$
- For all propositional formulae φ and ψ:

$$\begin{aligned}
\mathrm{len}(\neg\varphi) &:= 1 + \mathrm{len}(\varphi) \\
\mathrm{len}(\varphi \wedge \psi) &:= 1 + \mathrm{len}(\varphi) + \mathrm{len}(\psi) \\
\mathrm{len}(\varphi \vee \psi) &:= 1 + \mathrm{len}(\varphi) + \mathrm{len}(\psi)
\end{aligned}$$

Length of a Formula: Example

For $\varphi = \neg(p_1 \wedge \neg p_2)$, we have:

$$\begin{aligned}
\mathsf{len}(\varphi) &= 1 + \mathsf{len}(p_1 \wedge \neg p_2) \\
&= 1 + 1 + \mathsf{len}(p_1) + \mathsf{len}(\neg p_2) \\
&= 1 + 1 + 1 + 1 + \mathsf{len}(p_2) \\
&= 5
\end{aligned}$$

Proofs by Structural Induction

Goal: Prove a statement S about propositional formulae

By induction on the structure of propositional formulae:

1. Base Case:
   Prove S for all proposition symbols p_i.

2. Inductive Step:
   Prove that, if S holds for φ and ψ, then S also holds for:

$$\neg\varphi, \qquad \varphi \wedge \psi, \qquad \varphi \vee \psi$$

Proofs by Structural Induction: Example

Claim: For all propositional formulae φ, $|\mathsf{sub}(\varphi)| \leq \mathsf{len}(\varphi)$.

Proof: By induction on the structure of propositional formulae.

- Base case: For all integers $i \geq 0$,

$$|\mathsf{sub}(p_i)| = |\{p_i\}| = 1 = \mathsf{len}(p_i).$$

- Inductive step: Suppose the claim is true for φ and ψ. Then:

$$\begin{aligned}
|\mathsf{sub}(\varphi \wedge \psi)| &= |\{\varphi \wedge \psi\} \cup \mathsf{sub}(\varphi) \cup \mathsf{sub}(\psi)| \\
&\leq 1 + |\mathsf{sub}(\varphi)| + |\mathsf{sub}(\psi)| \\
&\leq 1 + \mathsf{len}(\varphi) + \mathsf{len}(\psi) \\
&= \mathsf{len}(\varphi \wedge \psi).
\end{aligned}$$

Exercise: $|\mathsf{sub}(\varphi \vee \psi)| \leq \mathsf{len}(\varphi \vee \psi)$ and $|\mathsf{sub}(\neg\varphi)| \leq \mathsf{len}(\neg\varphi)$

Termination of the Tableau Algorithm

Theorem

Let φ be a propositional formula. Then, the tableau algorithm terminates on input φ.

To prove the theorem, it suffices to show (why?):

There are only finitely many tableau paths starting in $\{\varphi\}$.

Notation

Definition

For every propositional formula φ, we define:

$$\mathrm{sub}^{\neg}(\varphi) := \mathrm{sub}(\varphi) \cup \{\neg\psi \mid \psi \in \mathrm{sub}(\varphi)\}.$$

Example:

$$\mathrm{sub}^{\neg}(\, p \vee \neg q\,) = \{\, p \vee \neg q,\ \neg(p \vee \neg q),\ p,\ \neg p,\ q,\ \neg q,\ \neg\neg q\,\}$$

Termination of the Tableau Algorithm (cont)

Claim: There are only finitely many tableau paths starting in $\{\varphi\}$.

Proof:

- Consider a tableau path $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_n$ with $\mathcal{C}_0 = \{\varphi\}$.
- Properties of the path:
  1. $\mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \cdots \subseteq \mathcal{C}_n \subseteq \mathsf{sub}^\neg(\varphi)$
  2. $|\mathcal{C}_0| < |\mathcal{C}_1| < \cdots < |\mathcal{C}_n| \leq |\mathsf{sub}^\neg(\varphi)|$

  Since $|\mathcal{C}_0| = 1$, property 2 implies $n < |\mathsf{sub}^\neg(\varphi)|$.
- Thus, each tableau path starting in $\{\varphi\}$ is a sequence of at most $|\mathsf{sub}^\neg(\varphi)|$ subsets of $\mathsf{sub}^\neg(\varphi)$.
- The number of such sequences is at most $|\mathsf{sub}^\neg(\varphi)|^{|\mathsf{sub}^\neg(\varphi)|}$.

Analysis of the Tableau Algorithm

Theorem

Let φ be a propositional formula. The following properties hold:

1. *Termination: The tableau algorithm terminates on input φ.* ✓
2. *Soundness: If the algorithms outputs "satisfiable", then φ is satisfiable.*
3. *Completeness: If φ is satisfiable, then the algorithm outputs "satisfiable".*

Soundness of the Tableau Algorithm

Theorem

Let φ be a propositional formula. If the tableau algorithm outputs "satisfiable" on input φ, then φ is satisfiable.

Proof: Let the tableau algorithm output "satisfiable" on input φ. Then, there is a complete tableau path $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_n$ such that $\mathcal{C}_n$ contains no clash.

Define an interpretation $\mathcal{I}$ such that for all integers $i \geq 1$:

$$\mathcal{I}(p_i) = \begin{cases} 1, & \text{if } p_i \in \mathcal{C}_n \\ 0, & \text{otherwise.} \end{cases}$$

Claim: $\mathcal{I}(\psi) = 1$ for all $\psi \in \mathcal{C}_n$

(Since $\varphi \in \mathcal{C}_n$, this implies that φ is satisfiable.)

Base Case: Proposition Symbols

For all proposition symbols p_i:

1. If $p_i \in \mathcal{C}_n$, then $\mathcal{I}(p_i) = 1$.
2. If $\neg p_i \in \mathcal{C}_n$, then $\mathcal{I}(p_i) = 0$.

Proof: Let p_i be a proposition symbol.

- If $p_i \in \mathcal{C}_n$, then $\mathcal{I}(p_i) = 1$ by the definition of $\mathcal{I}$.
- Suppose that $\neg p_i \in \mathcal{C}_n$. Since $\mathcal{C}_n$ contains no clash, we have $p_i \notin \mathcal{C}_n$. Hence, $\mathcal{I}(p_i) = 0$ by the definition of $\mathcal{I}$. $\square$

Inductive Step: Negation

Suppose that the claim is true for ψ. Then:

1. If $\neg\psi \in \mathcal{C}_n$, then $\mathcal{I}(\neg\psi) = 1$.
2. If $\neg\neg\psi \in \mathcal{C}_n$, then $\mathcal{I}(\neg\psi) = 0$.

Proof: We consider the two cases separately:

1. $\neg\psi \in \mathcal{C}_n$: Since the claim holds for ψ, this implies $\mathcal{I}(\psi) = 0$, and hence $\mathcal{I}(\neg\psi) = 1$.
2. $\neg\neg\psi \in \mathcal{C}_n$: Since the $\neg\neg$-rule is not applicable to $\mathcal{C}_n$ (otherwise, the tableau path $\mathcal{C}_0, \ldots, \mathcal{C}_n$ wouldn't be complete), we have $\psi \in \mathcal{C}_n$. This implies $\mathcal{I}(\psi) = 1$ (since the claim holds for ψ), and hence $\mathcal{I}(\neg\psi) = 0$. $\square$

Inductive Step: Conjunction

Suppose that the claim is true for ψ_1 and ψ_2. Then:

1. If $\psi_1 \wedge \psi_2 \in \mathcal{C}_n$, then $\mathcal{I}(\psi_1 \wedge \psi_2) = 1$.
2. If $\neg(\psi_1 \wedge \psi_2) \in \mathcal{C}_n$, then $\mathcal{I}(\psi_1 \wedge \psi_2) = 0$.

Proof: We consider the two cases separately:

1. $\psi_1 \wedge \psi_2 \in \mathcal{C}_n$: Since the $\wedge$-rule is not applicable to $\mathcal{C}_n$, we have $\psi_1 \in \mathcal{C}_n$ and $\psi_2 \in \mathcal{C}_n$. This implies $\mathcal{I}(\psi_1) = \mathcal{I}(\psi_2) = 1$ (as the claim holds for ψ_1 and ψ_2). Hence, $\mathcal{I}(\psi_1 \wedge \psi_2) = 1$.

2. $\neg(\psi_1 \wedge \psi_2) \in \mathcal{C}_n$: Since the $\neg\wedge$-rule is not applicable to $\mathcal{C}_n$, we have $\neg\psi_1 \in \mathcal{C}_n$ or $\neg\psi_2 \in \mathcal{C}_n$. This implies $\mathcal{I}(\psi_1) = 0$ or $\mathcal{I}(\psi_2) = 0$. Hence, $\mathcal{I}(\psi_1 \wedge \psi_2) = 0$. $\square$

Inductive Step: Disjunction

Suppose that the claim is true for ψ_1 and ψ_2. Then:

1. If $\psi_1 \vee \psi_2 \in \mathcal{C}_n$, then $\mathcal{I}(\psi_1 \vee \psi_2) = 1$.
2. If $\neg(\psi_1 \vee \psi_2) \in \mathcal{C}_n$, then $\mathcal{I}(\psi_1 \vee \psi_2) = 0$.

Proof: Analogous to the case of conjunctions (exercise). □

Analysis of the Tableau Algorithm

Theorem

Let φ be a propositional formula. The following properties hold:

1. *Termination: The tableau algorithm terminates on input φ.* ✓
2. *Soundness: If the algorithms outputs "satisfiable", then φ is satisfiable.* ✓
3. *Completeness: If φ is satisfiable, then the algorithm outputs "satisfiable".*

Completeness of the Tableau Algorithm

Theorem

Let φ be a propositional formula. If φ is satisfiable, then the tableau algorithm outputs "satisfiable" on input φ.

Proof: Let φ be satisfiable.

- Since φ is satisfiable, there is an interpretation $\mathcal{I}$ with $\mathcal{I}(\varphi)=1$.
- Guided by $\mathcal{I}$, we construct a complete tableau path $\mathcal{C}_0, \ldots, \mathcal{C}_n$ such that $\mathcal{C}_0 = \{\varphi\}$ and $\mathcal{C}_n$ contains no clash.
- Such a tableau path implies that the tableau algorithm outputs "satisfiable" on input φ.

Construction of the Tableau Path

The path $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_n$ is constructed inductively:

1. $\mathcal{C}_0 := \{\varphi\}$
2. Assume that $\mathcal{C}_i$ has been defined, and $\mathcal{I}(\psi) = 1$ for all $\psi \in \mathcal{C}_i$.
   (Note: $\mathcal{C}_i$ contains no clash.)

   Case 1: No completion rule is applicable to $\mathcal{C}_i$.
   Then, $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_n$ is the desired path.

   Case 2: Some completion rule is applicable to $\mathcal{C}_i$.
   Then, let $\mathcal{C}_{i+1}$ be the result of applying one of the completion rules to $\mathcal{C}_i$ such that $\mathcal{I}(\psi) = 1$ for all $\psi \in \mathcal{C}_{i+1}$.

Construction of C_{i+1}

Case 1: The $\wedge$-rule is applicable to $\mathcal{C}_i$.

Then, $\mathcal{C}_i$ contains a formula $\psi_1 \wedge \psi_2$ with $\psi_1 \notin \mathcal{C}_i$ or $\psi_2 \notin \mathcal{C}_i$.

- Since $\mathcal{I}(\psi_1 \wedge \psi_2) = 1$, we have $\mathcal{I}(\psi_1) = 1$ and $\mathcal{I}(\psi_2) = 1$.
- Define $\mathcal{C}_{i+1} := \mathcal{C}_i \cup \{\psi_1, \psi_2\}$.
- By construction: All formulae in $\mathcal{C}_{i+1}$ are true under $\mathcal{I}$.

Construction of $\mathcal{C}_{i+1}$

Case 2: The $\vee$-rule is applicable to $\mathcal{C}_i$.

Then, $\mathcal{C}_i$ contains a formula $\psi_1 \vee \psi_2$ with $\psi_1 \notin \mathcal{C}_i$ and $\psi_2 \notin \mathcal{C}_i$.

- Since $\mathcal{I}(\psi_1 \vee \psi_2) = 1$, we have $\mathcal{I}(\psi_1) = 1$ or $\mathcal{I}(\psi_2) = 1$.
- If $\mathcal{I}(\psi_1) = 1$, define $\mathcal{C}_{i+1} := \mathcal{C}_i \cup \{\psi_1\}$.
- If $\mathcal{I}(\psi_2) = 1$, define $\mathcal{C}_{i+1} := \mathcal{C}_i \cup \{\psi_2\}$.
- By construction: All formulae in $\mathcal{C}_{i+1}$ are true under $\mathcal{I}$.

Remaining completion rules: exercise.

Analysis of the Tableau Algorithm

Theorem

Let φ be a propositional formula. The following properties hold:

1. *Termination: The tableau algorithm terminates on input φ.* ✓
2. *Soundness: If the algorithms outputs "satisfiable", then φ is satisfiable.* ✓
3. *Completeness: If φ is satisfiable, then the algorithm outputs "satisfiable".* ✓