Propositional Calculus - *Semantics* (3/3)

Moonzoo Kim
CS Division of EECS Dept.
KAIST

moonzoo@cs.kaist.ac.kr
http://pswlab.kaist.ac.kr/courses/cs402-07

Overview

- 2.1 Boolean operators
- 2.2 Propositional formulas
- 2.3 Interpretations
- 2.4 Logical Equivalence and substitution
- 2.5 Satisfiability, validity, and consequence
- 2.6 Semantic tableaux
- 2.7 Soundness and completeness

Semantic tableaux

- The method of **semantic tableaux** is a relatively efficient algorithm for deciding satisfiability in the propositional calculus.
  - Search systematically for a model.
  - If one is found, the formula is satisfiable; otherwise, it is unsatisfiable.
- This method is the main tool for proving general theorems about the calculus.

Semantic tableaux

Definition 2.43

- A **literal** is an atom or a negation of an atom.
- An atom is a positive literal and the negation of an atom is a negative literal.
- For any atom p, $\{p, \neg p\}$ is a complementary pair of literals.
- For any formula A, $\{A, \neg A\}$ is a complementary pair of formulas.
- A is the complement of $\neg A$ and $\neg A$ is the complement of A.

Semantic tableaux

- Analyze the satisfiablity of $A = p \wedge (\neg q \vee \neg p)$

  $\nu(A) = T$ iff both $\nu(p) = T$ and $\nu(\neg q \vee \neg p) = T$.

  Hence, $\nu(A) = T$ if and only if either:

  1. $\nu(p) = T$ and $\nu(\neg q) = T$ or
  2. $\nu(p) = T$ and $\nu(\neg p) = T$

  ➔ {p, ¬ p} or {p, ¬ q}

Reduce the question of the satisfiability of formula *A* to question about the satisfiability of sets of literals. (top-down approach)

Semantic tableaux

- Formula $B = (p \vee q) \wedge (\neg p \wedge \neg q)$.
- $\nu(B) = T$ iff $\nu(p \vee q) = T$ and $\nu(\neg p \wedge \neg q) = T$.
- Hence, $\nu(B) = T$ iff $\nu(p \vee q) = \nu(\neg p) = \nu(\neg q) = T$.
- Hence, $\nu(B) = T$ iff either
  1. $\nu(p) = \nu(\neg p) = \nu(\neg q) = T$, or
  2. $\nu(q) = \nu(\neg p) = \nu(\neg q) = T$.

Since both $\{p, \neg p, \neg q\}$ and $\{q, \neg p, \neg q\}$ contain complementary pairs, B is unsatisfiable.

Semantic tableaux

- The systematic search is easy to conduct if a data structure is used to keep track of the assignments that must be made to subformulas.
- In semantic tableaux, *trees* are used.
- A leaf containing a complementary set of literals will marked $\times$, while a satisfiable leaf will be marked $\odot$.

Semantic tableaux

$p \wedge (\neg q \vee \neg p)$

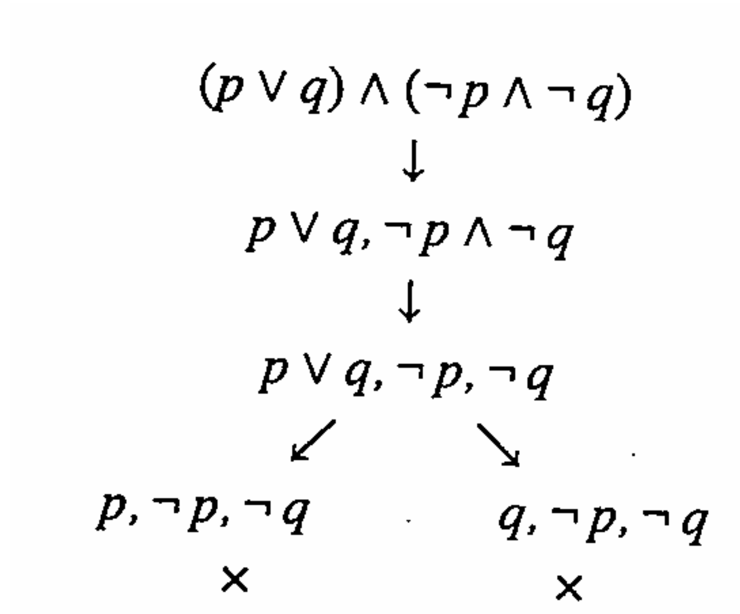
$\downarrow$

$p\ ,\quad \neg q \vee \neg p$

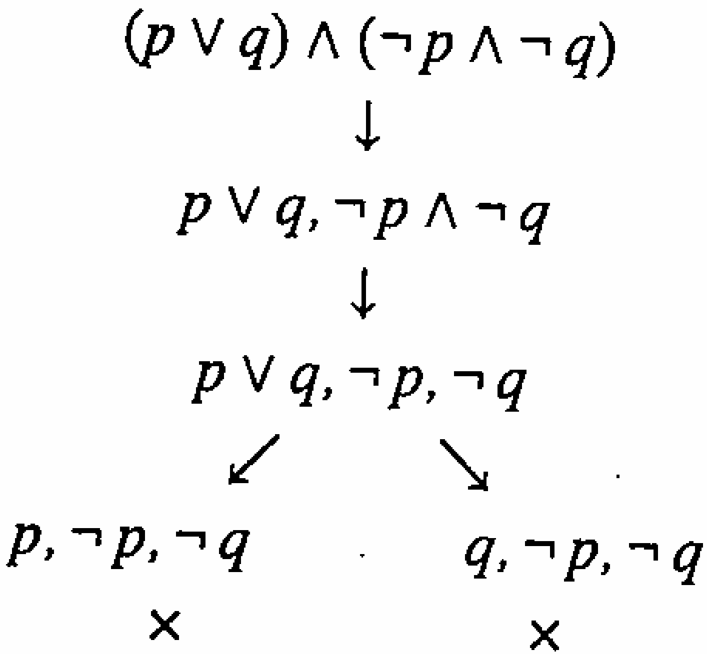
$\swarrow \qquad \searrow$

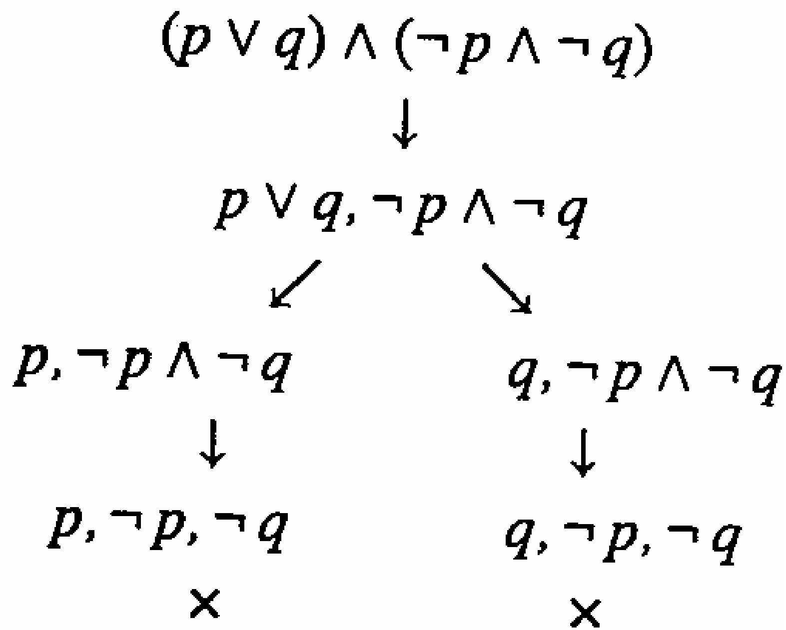
$p, \neg q \qquad\qquad p, \neg p$

$\odot \qquad\qquad \times$



Semantic tableaux





Semantic tableaux

- α–formulas are conjuctive and are satisfiable only if both subformulas α_1 and α_2 are satisfied
- β–formulas are disjuctive and are satisfied even if only one of the subformulas β_1 or β_2 is satisfiable.

α	α_1	α_2
$\neg\neg A_1$	A_1	
$A_1 \wedge A_2$	A_1	A_2
$\neg(A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$\neg(A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$\neg(A_1 \uparrow A_2)$	A_1	A_2
$A_1 \downarrow A_2$	$\neg A_1$	$\neg A_2$
$A_1 \leftrightarrow A_2$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$
$\neg(A_1 \oplus A_2)$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$

β	β_1	β_2
$\neg(B_1 \wedge B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \vee B_2$	B_1	B_2
$B_1 \rightarrow B_2$	$\neg B_1$	B_2
$B_1 \uparrow B_2$	$\neg B_1$	$\neg B_2$
$\neg(B_1 \downarrow B_2)$	B_1	B_2
$\neg(B_1 \leftrightarrow B_2)$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$
$B_1 \oplus B_2$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$

Semantic tableaux

- Algorithm 2.46 (Construction of a semantic tableau)

Input: A formula A of the propositional calculus.

Output: A semantic tableau $\mathcal{T}$ for A all of whose leaves are marked.

$\mathcal{T}$ for A is a tree each node of which will be labeled with a set of formulas

U(l) : the set of formula of leaf l.

The construction terminates when all leaves are marked $\times$ or $\odot$.

Semantic tableaux

- If $U(l)$ is a set of literals, check if there is a complementary pair of literals in $U(l)$. If so, mark the leaf closed $\times$; if not, mark the leaf as open $\odot$.
- If $U(l)$ is not a set of literals, choose a formula in $U(l)$ which is not a literal.
  - If the formula is an α–formula, create a new node l' as a child of l and label l' with $U(l') = (U(l) - \{\alpha\}) \cup \{\alpha_1, \alpha_2\}$.
  - If the formula is a β–formula, create two new nodes l' and l'' as children of l. Label l' with $U(l') = (U(l) - \{\beta\}) \cup \{\beta_1\}$, and label l'' with $U(l'') = (U(l) - \{\beta\}) \cup \{\beta_2\}$.

Semantic tableaux

Definition 2.47

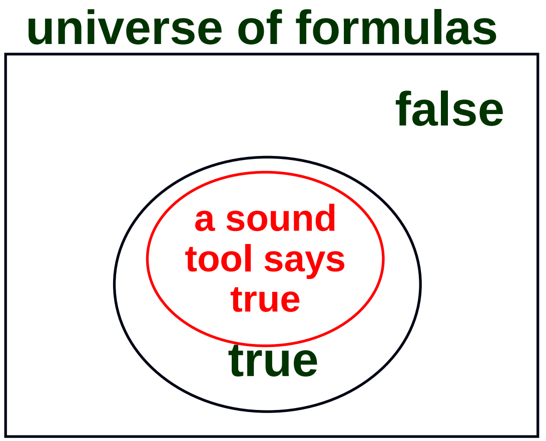
- A tableau whose construction has terminated is called a *completed tableau*.
- A completed tableau is **closed** if all leaves are marked closed (x). Otherwise, it is **open**.

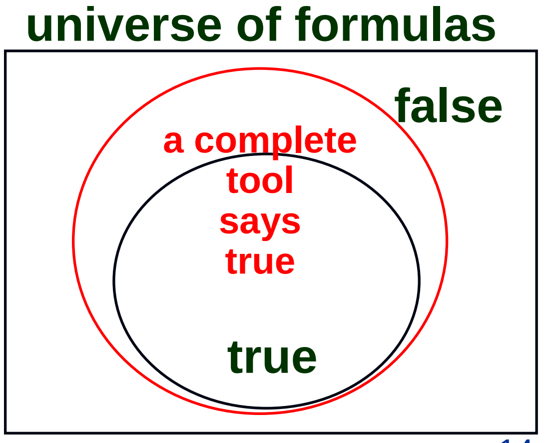
Theorem 2.48

- The construction of a semantic tableau terminates.

Soundness and completeness

- A tool is sound if the tool says that a formula ϕ is valid (validity, not satisfiability), then ϕ is really valid
- A tool is complete if ϕ is valid, then the tool says that ϕ is valid
  - Writing in a counter positive way
    - A tool (or method) is complete if the tool says that ϕ is not valid, then ϕ is really not valid
- Therefore, if a tool is sound and complete, then
  - the tool says that ϕ is valid iff ϕ is really valid
- Note that
  - if a dumb tool always says that ϕ is not valid, then that tool is still sound
  - if a dumb tool always says that ϕ is valid, then that tool is still complete





Soundness and completeness

Theorem 2.49(Soundness and completeness)

- Let $\mathcal{T}$ be a completed tableau for a formula A. A is unsatisfiable if and only if $\mathcal{T}$ is closed.

- Corollary 2.50 A is satisfiable if and only if $\mathcal{T}$ is open.
- Corollary 2.51 A is valid iff the tableau for $\neg A$ closes.
- Corollary 2.52 The method of semantic tableaux is a decision procedure for validity in the propositional calculus.

Soundness

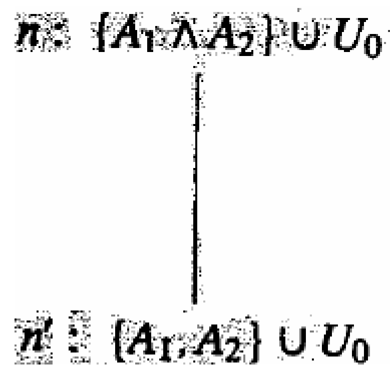
Proof of soundness

- if the tableau $\mathcal{T}$ for a formula *A* closes, then *A* is unsatisfiable.
- if a subtree rooted at node n of $\mathcal{T}$ closes, then the set of formulas *U*(*n*) labeling n is unsatisfiable.
  - *h*: height of the node n in $\mathcal{T}$.
  - If $h = 0$, *n* is a leaf. Since $\mathcal{T}$ closes, *U*(*n*) contains a complementary set of literals. Hence *U*(*n*) is unsatisfiable.

Soundness

- If $h > 0$, then some α - or β – rule was used in creating the child(ren) of n:
  - Case 1: An α –rule was used. $U(n) = \{A_1 \wedge A_2\} \cup U_0$ and $U(n') = \{A_1, A_2\} \cup U_0$ for some set of formulas U_0. But the height of n' is h–1, so $U(n')$ is unsatisfiable since the subtree rooted at n' closes. Let ν be an arbitrary interpretation. Since U(n') is unsatisfiable, $\nu(A') = F$ for some $A' \in U(n')$. There are three possiblities:
    - For some $A_0 \in U_0$, $\nu(A_0) = F$. But $A_0 \in U_0 \subseteq U(n)$.
    - $\nu(A_1) = F$. $\nu(A_1 \wedge A_2) = F$, and $A_1 \wedge A_2 \in U(n)$.
    - $\nu(A_2) = F$. $\nu(A_1 \wedge A_2) = F$, and $A_1 \wedge A_2 \in U(n)$.

    Thus $\nu(A) = F$ for some $A \in U(n)$; $U(n)$ is unsatisfiable.



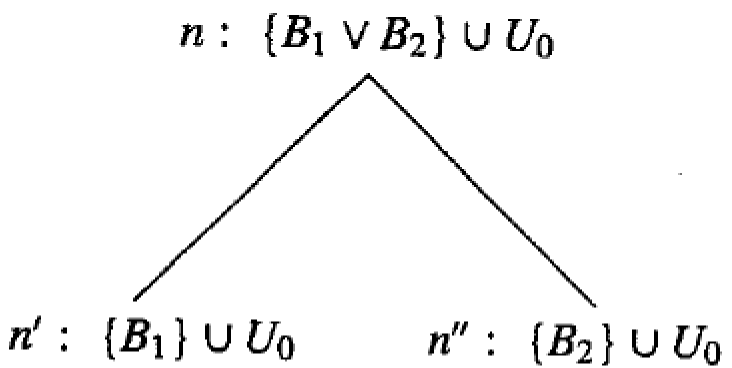
Soundness

- Case 2:

  A β –rule was used. $U(n) = \{B_1 \vee B_2\} \cup U_0$, $U(n') = \{B_1\} \cup U_0$, and $U(n'') = \{B_2\} \cup U_0$. By the inductive hypothesis, both $U(n')$ and $U(n'')$ are unsatisfiable. Let ν be an arbitrary interpretation. There are two possibilities:

  - $U(n')$ and $U(n'')$ are unsatisfiable because $\nu(B_0) = F$ for some $B_0 \in U_0$. But $B_0 \in U_0 \subseteq U(n)$.
  - Otherwise, $\nu(B_0) = T$ for all $B_0 \in U_0$. Since both $U(n')$ and $U(n'')$ are unsatisfiable, $\nu(B_1) = \nu(B_2) = F$. By definition of ν on $\vee$, $\nu(B_1 \vee B_2) = F$, and $B_1 \vee B_2 \in U(n)$.

  Thus $\nu(B) = F$ for some $B \in U(n)$; since ν was arbitrary, $U(n)$ is unsatisfiable.



Completeness

Proof of completeness

- If A is unsatisfiable then every tableau for A closes.
- Contrapositive statement
  - If some tableau for A is open (i.e., if some tableau for A has an open branch), then the formula A is satisfiable.

Completeness

Definition 2.57

- Let U be a set of formulas. U is a Hintikka set iff:
  1. For all atoms p appearing in a formula of U, either $p \notin U$ or $\neg p \notin U$.
  2. If $\alpha \in$ U is an α–formula, then $\alpha_1 \in$ U and $\alpha_2 \in$ U.
  3. If $\beta \in$ U is a β–formula, then $\beta_1 \in$ U or $\beta_2 \in$ U.

Theorem 2.59

Let l be an open leaf in a completed tableau $\mathcal{T}$.

Let $U = \bigcup_i U(i)$, where i runs over the set of nodes on the branch from the root to l. Then U is a Hintikka set.

Completeness

Theorem 2.60(Hintikka's Lemma)

- Let U be a Hintikka set. Then U is satisfiable.

Proof of completeness:

- Let $\mathcal{T}$ be a completed open tableau for *A*. Then *U*, the union of the labels of the nodes on an open branch, is a Hintikka set by Theorem 2.59 and a model can be found for *U* by Theorem 2.60. Since *A* is the formula labeling the root, $A \in U$, so the interpretation is a model of *A*.