Propositional Logic: Semantics (3/3)

CS402, Spring 2016

Shin Yoo

Questions that haunt us from our last lecture:

- In Def. 2.30, is it okay towrite $\nu(A_1) \wedge \ldots \wedge \nu(A_n) = T$ instead of $\nu(A_1) = \ldots = \nu(A_n) = T$?

Questions that haunt us from our last lecture:

- In Def. 2.30, is it okay towrite $\nu(A_1) \wedge \ldots \wedge \nu(A_n) = T$ instead of $\nu(A_1) = \ldots = \nu(A_n) = T$?
  - The logic behind the question: if we assume that formulas are finite strings as well as that the set U in Def. 2.30 can be infinite, we end up in a contradiction where the conjunctive form of the satisfiability condition becomes an infinite formula (string), whereas the enumeration of individual satisfiability conditions results in an *infinite number* of *finite formulas*.

Questions that haunt us from our last lecture:

- In Def. 2.30, is it okay towrite $\nu(A_1) \wedge \ldots \wedge \nu(A_n) = T$ instead of $\nu(A_1) = \ldots = \nu(A_n) = T$?
  - The logic behind the question: if we assume that formulas are finite strings as well as that the set U in Def. 2.30 can be infinite, we end up in a contradiction where the conjunctive form of the satisfiability condition becomes an infinite formula (string), whereas the enumeration of individual satisfiability conditions results in an *infinite number* of *finite formulas*.
  - Good point! We donot deal with infinity *directly* in this course, it is a whole different can of worms. Let us accept that formulas are finite, U can be infinite, and the enumerative form of satisfiability condition is sufficient :)

Questions that haunt us from our last lecture:

- Why is it that the necessary condition for $\mathcal{T}(U)$ to be a theory is $U \models U$?

Questions that haunt us from our last lecture:

- Why is it that the necessary condition for $\mathcal{T}(U)$ to be a theory is $U \models U$?
  - The nature of the initial, original confusion still *somewhat* escapes me :(

Questions that haunt us from our last lecture:

- Why is it that the necessary condition for $\mathcal{T}(U)$ to be a theory is $U \models U$?
  - The nature of the initial, original confusion still *somewhat* escapes me :(
  - The critical part was the meaning of entailment: there are no such things as direct entailment and secondary or thirdly entailment.

Questions that haunt us from our last lecture:

- Why is it that the necessary condition for $\mathcal{T}(U)$ to be a theory is $U \models U$?
  - The nature of the initial, original confusion still *somewhat* escapes me :(
  - The critical part was the meaning of entailment: there are no such things as direct entailment and secondary or thirdly entailment.
  - That is, if $U \models A$ and $A \models B$, then $U \models B$. The confusion was resolved when it became clear that it is not necessary to show that B in our example belongs to $\mathcal{T}(U)$, as it is obviously implied by the definition of $\mathcal{T}(U)$.

Overview

- Semantic Tableaux
- Soundness and completeness

Semantic tableaux: 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.

Definition 1 (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

Definition 1 (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.

Important observation: a set of litrals is *satisfiable* if and only if it does **not** contain a *complementary* pair of literals.

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 $\nu(\neg p) = T$

$\therefore \{p, \neg p\}$ or $\{p, \neg q\}$.

In other words, the process is to reduce the question from one about the satisfiability of a formula to one about the satisfiability of sets of *literals*.

Since any formula contains *finite* atoms, there are at most *finite* number of sets of literals. Then the decision on satisfiability becomes trivial.

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

- This systematic search becomes easier if we use a suitable data structure 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 with a $\times$ symbol, while a leaf containing a satisfiable set of literals will be marked with a $\odot$ symbol.

Semantic tableaux

Is $p \wedge (\neg q \vee \neg p)$ satisfiable?

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

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

$$p, \neg q\ (\odot) \qquad p, \neg p\ (\times)$$

Is $(p \vee q) \wedge (\neg p \wedge \neg q)$ satisfiable?

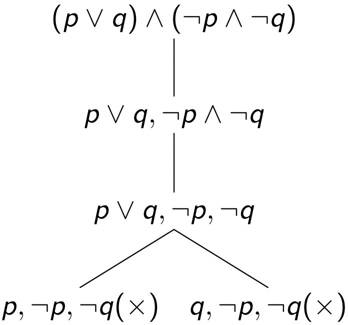
$$(p \vee q) \wedge (\neg p \wedge \neg q)$$

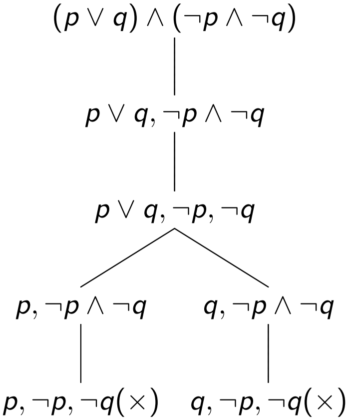
$$p \vee q, \neg p \wedge \neg q$$

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

$$p, \neg p, \neg q (\times) \qquad q, \neg p, \neg q (\times)$$

Is $(p \lor q) \land (\neg p \land \neg q)$ satisfiable?





Semantic tableaux

- α-formulas are conjunctive and are satisfiable only if both subformulas, α_1 and α_2, are satisfied.
- β-formulas are disjunctive and are satsified if at least 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

Let $\mathcal{T}$ for a propositional formula A be a tree, whose nodes are all labeled with a set of formulas. Let $U(l)$ be the set of formulas of leaf l.

CONSTRUCTION OF SEM. TAB.(Algorithm 2.46)
Input: A propositional formula A
Output: A semantic tableaux $\mathcal{T}$ for A with marked leaves

(1) $\mathcal{T} \leftarrow$ a tree with a single node labeled A
(2) **while** there exists an unmarked leaf
(3) **foreach** unmarked leaf l
(4) **if** $U(l)$ is a set of lit.
(5) **if** a compl. lit. pair $\in U(l)$ **then** Mark l as $\times$
(6) **else** Mark l as $\oplus$
(7) **else**
(8) Choose $A \in U(l)$
(9) **if** $A == \alpha$ **then** Add l' to l s.t. $U(l') \leftarrow (U(l) - \{\alpha\} \cup \{\alpha_1, \alpha_2\}$
(10) **if** $A == \beta$ **then** Add l', l'' to l s.t. $U(l') \leftarrow (U(l) - \{\beta\}) \cup \{\beta_1\}$, $U(l'') \leftarrow (U(l) - \{\beta\}) \cup \{\beta_2\}$

Definition 2 (2.47)

- A tableau whose construction has terminated is called a *completed tableau*.
- A completed tableau is *closed* if all leaves are marked closed (i.e. $\times$); otherwise, it is *open*.

Theorem 1 (2.48)

The construction of a semantic tableau terminates.

Soundness and Completeness

- A tool is *sound* if whenever the tool says that a formula ϕ is valid (validity, not satisfiability), ϕ is really valid. That is, $\vdash \phi$ implies $\models \phi$.
- A tool is complete if whenever ϕ is valid, the tool does say that ϕ is valid. That is, $\models \phi$ implies $\vdash \phi$.
  - Writing in a contra-positive way: a tool (or method) is complete if whenever 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 (2.49)

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

Corollary 1 (2.50)

A is satisfiable if and only if $\mathcal{T}$ is open.

Corollary 2 (2.51)

A is valid if and only if the tableau for $\neg A$ is closed.

Corollary 3 (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. Let h be the 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 $h > 0$, either α- or β- rule was used in creating the child(ren) of n:
  - Case 1: α-rule. $U(n) = \{A_1 \wedge A_2\} \cup U_0$ and $U(n') = \{A_1, A_2\} \cup U_0$ for some set of formulas U_0.
  - The height of n' is $h - 1$; by induction, $U(n')$ 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 possibilities:
    - 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 $A_1 \wedge A_2 \in U(n)$.

    In all three cases, $\nu(A) = F$ for some $A \in U(n)$. Therefore, $U(n)$ is unsatisfiable.

Soundness

- If $h > 0$, either α- or β- rule was used in creating the child(ren) of n:
  - Case 2: β-rule. $U(n) = \{B_1 \vee B_2\} \cup U_0$, $U(n) = \{B_1\} \cup U_0$ and $U(n'') = \{B_2\} \cup U_0$ for some set of formulas U_0.
  - By induction, both $U(n')$ and $U(n'')$ are unsatisfiable, since the subtrees rooted at n' and n'' close.
  - Let ν be an arbitrary interpretation. There are three 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 $B_1 \vee B_2 \in U(n)$.

  Therefore $\nu(B) = F$ 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 (Cor 2.50): 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 3 (2.57)

Let U be a set of formulas. U is a **Hintikka**[a] 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 an β-formula, then either $\beta_1 \in U$ or $\beta_2 \in U$.

[a]Named after Finnish logician Jaakko Hintikka (1929-2015).

Theorem 3 (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 4 (2.60)

Hintikka's Lemma: *Let U be a Hintikka set. Then U is satisfiable.*

Proof.

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. □