1 First-Order Consistency Properties

Remark 31.1.1 The method of semantic tableaux is a complete proof system because it was designed to establish its own completeness, in the sense that a *failed proof* leads to a construction of a Hintikka set formed from a finished non-contradictory path. More common among proof systems are those like the method of natural deduction, in which a failed proofs does not point in the direction of a counterexample, and may only mean a failure in our ingenuity.

In Lecture 16 we abstracted out the essence of the completeness proof for semantic tableaux for propositional logic by means of *Propositional Consistency Properties.* If $\mathcal{C}$ was a propositional consistency property (a collection of sets of propositions) and Γ is a set of propositions in $\mathcal{C}$, then Γ is satisfiable. The conditions of defining a propositional consistency properties replaced the tableaux rules in establishing Hintikka's Lemma 12.1.4.

We introduce here the notion of a *First-Order Consistency Property* which extends that of propositional consistency property. If $\mathcal{C}$ is a collection of sets of first-order sentences and is a first-order consistency property, then any $\Gamma \in \mathcal{C}$ is satisfiable.

Remark 31.1.2 (Completeness and consistency) Most completeness proofs argue by contraposition: if there is no proof of a sentence ϕ from a set of premises Γ, then there is a structure $\mathcal{A}$ which satisfies $\Gamma \cup \{\phi\}$. We generalize this observation by introducing the notion of *consistency.*

- Let $\mathscr{P}$ be a proof system (such as semantic tableaux or natural deduction). A set of sentences is $\mathscr{P}$*-consistent* if no contradiction can be derived using the proof system's machinery.

In the case of tableaux, we could directly use its machinery to test whether $\Gamma \cup \{\neg\, \phi\}$ was tableaux-consistent: if not, we have a proof (a contradictory tableau), and if so, we have a Hintikka set (a noncontradictory finished path) satisfying all the sentences in the set.

By looking carefully at such constructions, one can identify those features of consistency which are essential in ensuring a satisfying structure exists. An *abstract consistency property* is a property which captures essential features of a notion of consistency and the *Model Existence Theorem* is the assertion that these features are sufficient to guarantee the existence of a suitable structure (also called a *model*).

Instead of talking about a consistency *property* of sets of sentences, we will talk about the *collection* $\mathcal{C}$ of all sets which are consistent (relative to whatever notion of consistency we are studying). We are identifying the collection of consistent sets of sentences with the property of being consistent. An abstract consistency property is defined to be a collection $\mathcal{C}$ of sets of sentences which meet certain closure conditions, very similar to the conditions used to identify Hintikka sets.

Definition 31.1.3 (First-Order Consistency Property) Let $\mathcal{L}$ be a first-order language and A be a set of parameters not in $\mathcal{L}$, and let $\mathcal{C}$ be a collection of sets of sentences in the extended language $\mathcal{L}^A$. We call $\mathcal{C}$ a *first-order consistency consistency property*, and the sets $S \in \mathcal{C}$ $\mathcal{C}$-consistent, if it meets the following conditions for each $S \in \mathcal{C}$:

(C1) No atomic sentence and its negation are in S Also, $\perp \notin S$ and $\neg \top \notin S$.

(C2) If α is a type-A sentence with components α_1 and α_2 and $S \cup \{\alpha\}$ is $\mathcal{C}$-consistent, then so is $S \cup \{\alpha_1, \alpha_2\}$.

(C3) If β is a type-B sentence with components β_1 and β_2, and $S \cup \{\beta\}$ is $\mathcal{C}$-consistent, then *at least one of* $S \cup \{\beta_1\}$ or $S \cup \{\beta_1\}$ is $\mathcal{C}$-consistent.

(C4) If γ is a type-C sentence and $S \cup \{\gamma\}$ is $\mathcal{C}$-consistent, then $S \cup \{\gamma(t)\}$ is $\mathcal{C}$-consistent for each variable-free term t in the extended language $\mathcal{L}^A$.

(C5) If δ is a type-D sentence and $S \cup \{\delta\}$ is $\mathcal{C}$-consistent, then $S \cup \{\delta(a)\}$ is $\mathcal{C}$-consistent for each parameter a not occurring in any sentence of S or δ.

Example 31.1.4 Let $\mathcal{C}$ be the collection of all satisfiable sets of sentences. Then $\mathcal{C}$ is a first-order consistency property:

(C1) No satisfiable set of sentences S can contain an atomic sentence and its negation nor $\bot$ nor $\neg\top$.

(C2) If $S \cup \{\alpha\}$ is satisfiable and α is a type-A sentence with components α_1 and α_2, then $S \cup \{\alpha_1, \alpha_2\}$ is satisfiable, because $\alpha \leftrightarrow (\alpha_1 \wedge \alpha_2)$ is valid. (So, if $v(\alpha) = \mathbf{T}$, then $v(\alpha_1) = \mathbf{T}$ and $v(\alpha_2) = \mathbf{T}$.)

(C3) If $S \cup \{\beta\}$ is satisfiable and βS is a type-B sentence with components β_1 and β_2, then *at least one of* $S \cup \{\beta_1\}$ or $S \cup \{\beta_1\}$ is satisfiable, because $\beta \leftrightarrow (\beta_1 \vee \beta_2)$ is valid. (So, if $v(\beta) = \mathbf{T}$, at least one of $v(\beta_1) = \mathbf{T}$ or $v(\beta_2) = \mathbf{T}$.)

(C4) If $S \cup \{\gamma\}$ is satisfiable and γ is a type-C sentence then $S \cup \{\gamma(t)\}$ is satisfiable because $\gamma \leftrightarrow (\forall x)\gamma(x)$ is valid. (If $\mathcal{A}$ is any structure satisfying $S \cup \{\gamma\}$, extend $\mathcal{A}$ by letting $t^{\mathcal{A}}$ be any element of the domain A, if t is uninterpreted, then $S \cup \{\gamma(t)\}$ will be satisfied in $\mathcal{A}^*$.)

(C5) If $S \cup \{\delta\}$ is satisfiable and δ is a type-D sentence then $S \cup \{\delta(c)\}$ is satisfiable, where c is a parameter new to S and δ because $\delta \leftrightarrow (\exists x)\delta(x)$ is valid. (If $\mathcal{A}$ is any structure satisfying $S \cup \{\delta\}$, then $\delta(a)$ will be true $\mathcal{A}$, so extend this structure to $\mathcal{A}^*$ where $c^{\mathcal{A}^*} = a$. Then $S \cup \{\delta(c)\}$ will be satisfied will be satisfied in $\mathcal{A}^*$.)

When working with proof systems it is usually more convenient to work with *inconsistency* rather than *consistency*:

Definition 31.1.5 (First-Order Inconsistency Property) Let $\mathcal{L}$ be a first-order language and A be a set of parameters not in $\mathcal{L}$, and let $\mathcal{I}$ be a collection of sets of sentences in the extended language $\mathcal{L}^A$. We call $\mathcal{I}$ a *first-order inconsistency consistency property*, and the sets $S \in \mathcal{C}$ $\mathcal{I}$-inconsistent, if it meets the following conditions for each $S \in \mathcal{I}$:

(I1) Any set S with an atomic sentence and its negation, or $\bot \in S$ or $\neg\top \in S$ is $\mathcal{I}$-inconsistent.

(I2) If α is a type-A sentence with components α_1 and α_2 and $S \cup \{\alpha_1, \alpha_2\}$ is $\mathcal{I}$-inconsistent then so is $S \cup \{\alpha\}$.

(I3) If β is a type-B sentence with components β_1 and β_2, and both $S \cup \{\beta_1\}$ and $S \cup \{\beta_2\}$ are $\mathcal{I}$-inconsistent, then so is $S \cup \{\beta\}$.

(I4) If γ is a type-C sentence and $S \cup \{\gamma(t)\}$ is $\mathcal{I}$-inconsistent, where t is a term in the extended language, then $S \cup \{\gamma\}$ is $\mathcal{I}$-inconsistent .

(I5) If δ is a type-D sentence and $S \cup \{\delta(a)\}$ is $\mathcal{I}$-inconsistent, where a is a parameter not occurring in S or δ, then $S \cup \{\delta\}$ is $\mathcal{I}$-inconsistent

Example 31.1.6 The collection of *unsatisfiable* sets of sentences is a first-order inconsistency property, as you can verify. The collection of all tableau-inconsistent sets of sentences is a first-order l inconsistency property as well. Recall that a set S is tableau-inconsistent if there is a contradictory tableau for S.

(I1) If S contains some atomic sentence and its negation or $\bot$ or $\neg\top$ then S is tableau-inconsistent.

(I2) Suppose $S \cup \{\alpha_1, \alpha_2\}$ is tableau inconsistent with contradictory tableau τ. This tableau may introduce α_1 and α_2 as part of the introduction rule from premises. The following will be a contradictory tableau for $S \cup \{\alpha\}$:

$$\begin{array}{c} \alpha \\ \tau' \end{array}$$

where τ' is obtained from τ by replacing all occurrences of α_1 and α_2 with the pair of α_1 followed by α_2, so that τ' is also contradictory. Each occurrence of α_1 and α_2 in τ' is justified by the type-A rule.

(I3) Suppose both $S \cup \{\beta_1\}$ and $S \cup \{\beta_2\}$ are tableau-inconsistent. Let τ_1 be a contradictory tableau for the first set and τ_2 be a contradictory tableau for the second set. Furthermore, we may assume that β_1 occurs at the root of τ_1 and β_2 occurs at the root of τ_2 (the order of introducing sentences is irrelevant). Then the following is a contradictory tableau for $S \cup \{\beta\}$:

$$\begin{array}{ccc} & \beta & \\ \beta_1 & & \beta_2 \\ \tau_1' & & \tau_2' \end{array}$$

where τ_1' is obtained from τ_1 by removing β_1 from the root, and similarly for τ_2'.

(I4) Suppose $S \cup \{\gamma(t)\}$ is tableau-inconsistent and let τ be a contradictory tableau. Then the following is a contradictory tableau for $S \cup \{\gamma\}$

$$\begin{array}{c} \gamma \\ \tau' \end{array}$$

where each occurrence of $\gamma(t)$ is replaced by $\gamma, \gamma(t)$ in τ to produce τ'. The justification for this replacement is the type-C rule.

(I5) Suppose $S \cup \{\delta(a)\}$ is tableau-inconsistent for each parameter a not occurring in S or in δ and let τ_a be a tableau establishing this (each such parameter a may involve a different tableaux. Furthermore, we will assume that the parameter a is never used in a type-D inference in τ_a. Then the following is contradictory tableau for $S \cup \{\delta\}$

$$\begin{array}{c} \delta \\ \delta(a) \\ \tau_a \end{array}$$

The connection between first-order consistency and inconsistency properties is brought out in the following:

Lemma 31.1.7 If $\mathcal{C}$ be a first-order consistency property, then the collection $\mathcal{I}$ of all sets which are *not* $\mathcal{C}$-consistent is a first-order inconsistency property.

Conversely, if $\mathcal{I}$ be a first-order inconsistency property, then the collection $\mathcal{C}$ of all sets which are *not* $\mathcal{I}$-inconsistent is a first-order consistency property.

Proof. Let $\mathcal{C}$ be a first-order consistency property and $\mathcal{I}$ the collection of all sets which are *not* $\mathcal{C}$-consistent. By (C1) no set S containing an atomic sentence and its negation, or $\perp$ or $\neg\top$ can be $\mathcal{C}$-consistent, so that any such set S is $\mathcal{I}$-inconsistent. This establishes (I1).

For (I2): If $S \cup \{\alpha_1, \alpha_2\}$ is $\mathcal{I}$-inconsistent, then $S \cup \{\alpha\}$ is not $\mathcal{C}$-consistent, since otherwise by (C2) $S \cup \{\alpha_1, \alpha_2\}$ would be $\mathcal{C}$-consistent.

For (I3): if both $S \cup \{\beta_1\}$ and $S \cup \{\beta_2\}$ are $\mathcal{I}$-inconsistent, then so is $S \cup \{\beta\}$, since otherwise by (C3) at least one of $S \cup \{\beta_1\}$ or $S \cup \{\beta_2\}$ would be $\mathcal{C}$-consistent.

For (I4): if $S \cup \{\gamma(t)\}$ is $\mathcal{I}$-inconsistent for some term t, then $S \cup \{\gamma\}$ cannot be $\mathcal{C}$-consistent by condition (C4).

For (I5): if $S \cup \{\delta(a)\}$ is $\mathcal{I}$-inconsistent for some parameter a not occurring in S or δ, then $S \cup \{\delta\}$ cannot be $\mathcal{C}$-consistent by condition (C5).

The converse claim is proven similarly. □

2 Model Completeness Theorem

This section is dedicated to our main result:

Theorem 31.2.1 (Model Existence Theorem) If $\mathcal{C}$ is a first-order consistency property and S is $\mathcal{C}$-consistent, then S is satisfiable.

Proof. Let $\mathcal{C}$ be a propositional consistency property and $S \in \mathcal{C}$. We will show that S can be extended to a Hintikka set H. It then follows by Hintikka's Lemma 29.3.2 that H is satisfiable, so that S is as well.

The construction of H is analogous to the construction of a finished tableau for S, except that we must use the properties (C1) to (C5) instead of the rules for constructing tableaux. Let $S = \phi_0, \phi_1, \phi_2, \ldots$ be an enumeration of S. We will build a sequence of sets $H_0, H_1, \ldots$ and let $H = \bigcup_n H_n$ so that H is a Hintikka set and $S \subseteq H$. The key is that each H_n will be $\mathcal{C}$-consistent. The construction is in stages n. We will also keep track of a set T of sentences that we have added to S to fulfill the Hintikka conditions. The construction will add only finitely many sentences to S at each stage n. We will also take an infinite set of new parameters $A = \{a_1, a_2, \ldots\}$, so that none of these parameters occur in S. Let $t_1, t_2, \ldots$ be the set of all variable-free terms in the original language together with the new parameters A.

Stage $n = 0$. Let $H_0 = S$. Note that H_0 is $\mathcal{C}$-consistent, since S is. We also let $T = \langle \phi_0 \rangle$, so that each sentence on T is also in H_0.

Stage $n+1$. Suppose we have constructed H_n which is $\mathcal{C}$-consistent and a finite sequence of sentences T each of which is in H_n. Let $T = \langle \psi_1, \psi_2, \ldots, \psi_m \rangle$ be the sentences in T. There are two steps. Step 1, extend $T = \langle \psi_1, \psi_2, \ldots, \psi_m, \phi_{n+1} \rangle$ by adding ϕ_{n+1} to the end of T. Step 2, let ψ_j be the least index of a sentence which has not yet been reduced and proceed according to its type.

1. If ψ_j is atomic, do nothing.
2. If $\psi_j = \alpha$ is type-A with components α_1 and α_2, and let $H_{n+1} = H_n \cup \{\alpha_1, \alpha_2\}$ and extend T to include α_1 and α_2. It follows by (C2) that H_{n+1} is $\mathcal{C}$-consistent.
3. If $\psi_j = \beta$ is type-B with components, β_1 and β_2 let

$$H_{n+1} = H_n \cup \begin{cases} \{\beta_1\} & \text{if } H_n \cup \{\beta_1\} \text{ is } \mathcal{C}\text{-consistent,} \\ \{\beta_2\} & \text{otherwise.} \end{cases}$$

   Extend T to include whichever of β_1 or β_2 were added to H_{n+1}.. It follows by (C3) that H_{n+1} is $\mathcal{C}$-consistent.
4. If $\psi_j = \gamma$ is type-C, then let t_i be the least indexed term such that $\gamma(t)$ is not yet in H_n. Let $H_{n+1} = H_n \cup \{\gamma(t)\}$ and extend T to include $\gamma(t)$ and γ again. It follows by (C4) that H_{n+1} is $\mathcal{C}$-consistent.
5. If $\psi_j = \delta$ is type-D, then let a be some parameter not yet occurring in H_n; there must be such parameters, since no parameter occurs in S, so the only occurrences in H_n must have come from sentences also placed on T; since T is finite, only finitely many parameters occur in H_n. Let $H_{n+1} = H_n \cup \{\delta(a)\}$ and add $\delta(a)$ to T¿ It follows by (C5) that H_{n+1} is $\mathcal{C}$-consistent.

This concludes step 2 and stage $n+1$. We have added at most 3 sentences to T, so T is finite and H_{n+1} is $\mathcal{C}$-consistent.

We have constructed a sequence of sets $H_0, H_1, H_2 \ldots$. Let $H = \bigcup_n H_n$. So, $S \subseteq H$. We will show that H is a Hintikka set by showing it satisfies properties (H0)-(H4) from Definition 29.3.2. Note that every sentence in H is eventually put onto the list T. If $\psi \in S$, then $\psi = \phi_i$ for some i, so at stage i ϕ_i is put onto T. All other sentences are put onto T simultaneously with their addition to H.

(H0). Suppose some atomic sentence A and its negation $\neg A$ are in H. Then at some large enough stage n, $A, \neg A \in H_n$. But H_n is $\mathcal{C}$-consistent, so this cannot be the case. Similarly, we cannot have $\perp$ or $\neg \top$ in H.

(H1). Suppose $\alpha \in H$ is a type-A sentence with components α_1 and α_2. Then α is on the list T, so there is a stage n in which $\alpha = \psi_n$ is to be reduced. At this point both components α_1 and α_2 were place into H.

(H2). Suppose β is a type-B sentence with components β_1 and β_2. Then β is on the list T, so there is a stage n in which $\beta = \psi_n$ is to be reduced. At this point one of the components β_1 or β_2 were place into H.

(H3). Suppose γ is a type-C sentence. Let $t = t_i$ be a term in the language. γ occurs on the list T, and at every stage it is reduced it is placed back onto the list. So, γ occurs at infinitely many positions on T. For large enough n, $\gamma(t_i)$ is added to H.

(H4). Suppose δ is a type-C sentence. Since δ is on the list T, there is a stage in which $\delta = \psi_n$ is reduced. At this stage $\delta(a)$ is added to H, where a is a parameter new to H.]]

So, $S \subseteq H$ and H is a Hintikka set. H is satisfiable by Hintikka's Lemma 29.3.2, and thus so is S.

$\square$

Remark 31.2.2 Although the construction of the Hintikka set in the Model Completeness Theorem did not appeal to the construction of a semantic tableau, you can view it as the construction of the left-most noncontradictory branch of a semantic tableau for the set S.

The following corollary is the contrapositive form of the Model Existence Theorem and follows from it by Lemma 31.1.7

Corollary 31.2.3 If $\mathcal{I}$ is a first-order inconsistency property and S is not $\mathcal{I}$-inconsistent, then S is satisfiable.

Remark 31.2.4 One easy application of the Model Existence Theorem is "another" proof of the Strong Completeness Theorem for semantic tableaux.

Corollary 31.2.5 (Strong Completeness Theorem for semantic tableaux) If Γ is tableau-consistent, then Γ is satisfiable.

Tableau-inconsistency is a propositional inconsistency property from Example 31.1.6 above, so that a set that is tableau-consistent is satisfiable by Corollary 31.2.3.