1 Hintikka Sets

Remark 12.1.1 Hintikka sets are named for the Finnish logician Jaako Hintikka, who introduced them. Note that the tableau rule correspond exactly to the Hintikka set conditions. At bottom the paths in a tableaux are simply attempts to construct Hintikka sets. Hintikka's lemma to follow makes this clearer.

Definition 12.1.2 A set of propositions S is a *Hintikka set* if the following conditions hold.

(H1) No propositional symbols and its negation are in S. Also, $\bot$ and $\neg\top$ are not in S

(H2) If a type-A proposition is in S, then both its components A_1 and A_2 are in S.

(H3) If type-B is in S, then at least one of its components B_1 or B_2 are in S.

Remark 12.1.3 Note that conditions (H2) and (H3) state if a certain type of proposition is in a Hintikka set, then so are some that are simpler (the components). Hintikka sets are also said to be *downward saturated* for this reason.

Hintikka's lemma extracts the essence of the method of tableaux, and plays a central role in the completeness theorem for the method of tableaux.

Lemma 12.1.4 (Hintikka's Lemma) Every Hintikka set is satisfiable.

Proof. Let S be a Hintikka set. A valuation v agrees with a proposition α i We will define an assignment v from S as follows: for each propositional symbol P in S let $\mathcal{A}(P) = \mathbf{T}$; for each propositional symbol P not in S, let $\mathcal{A}(P) = \mathbf{F}$. Let $v_{\mathcal{A}}$ be the valuation extending $\mathcal{A}$. We will show that if $\alpha \in S$ then $v_{\mathcal{A}}(\alpha) = \mathbf{T}$. The proof is by induction using unified notation.

By condition (H1) no proposition symbol P and its negation $\neg P$ are both in S. So, $v_{\mathcal{A}}(P) = \mathcal{A}(P) = \mathbf{T}$ if and only if $P \in S$. If $\neg P \in S$, then $P \notin S$, so $v_{\mathcal{A}}(P) = \mathbf{F}$. Since $\bot$ is not in S, there is no problem that $v_{\mathcal{A}}(\bot) = \mathbf{F}$. Since $\neg\top$ is not in S, there is no problem that $v_{\mathcal{A}}(\neg\top) = \mathbf{T}$. So, if α is a atomic or the negation of an atomic proposition, if $\alpha \in S$, then $v_{\mathcal{A}}(\alpha) = \mathbf{T}$.

Suppose that a type-A proposition is in S, so that its components A_1 and A_2 are in S by (H2). Then $v(A_1) = \mathbf{T}$ and $v(A_2) = \mathbf{T}$, so that $v(A) = \mathbf{T}$, since $A \simeq A_1 \wedge A_2$.

Suppose that a type-B proposition is in S, so that at least one of its components B_1 or B_2 are in S by (H3). Then by the inductive hypothesis, at least one of $v(B_1) = \mathbf{T}$ or $v(B_2) = \mathbf{T}$, so that $v(B) = \mathbf{T}$, as $B \simeq B_1 \vee B_2$. □

2 Completeness Theorem for Semantic Tableaux

The next theorem states that the method of constructing tableaux is nothing more than the attempt to build a Hintikka set.

Theorem 12.2.1 If τ is a finished tableau and π a non-contradictory path through τ. Then π is a Hintikka set (that is, the collection of propositions on π is a Hintikka set). Thus, any non-contradictory finished path π is satisfiable.

Proof. Let π be non-contradictory finished path. Then there is no propositional symbol and its negation on π, nor $\bot$ nor $\neg\top$ on π, so that (H1) is satisfied. If type-A proposition is on π, then since π is finished both of its components A_1 and A_2 are on π, so π satisfies (H2). If a type-B proposition is on π, then since π is finished at least one of its components B_1 or B_2 are on π, so π satisfies (H3). Thus, π is a Hintikka set.

A non-contradictory finished path is a Hintikka set, and all Hintikka sets are satisfiable by Hintikka's Lemma 12.1.4, so any non-contradictory path is satisfiable. □

Theorem 12.2.2 (completeness theorem for semantic tableaux) If α is a tautology, then there is a tableau deduction of α, i.e.

$$\models \alpha \quad \text{implies} \quad \vdash \alpha$$

Proof. The proof is by contraposition. We suppose that there is no tableau proof of α, and show that $\neg\alpha$ is satisfiable (so that α cannot be a tautology). By Theorem 8.3.5 there is a finished tableau τ with $\neg\alpha$ at its root. (In fact, there is a finite finished tableau, but this is not used here). Since there is no contradictory finished tableau for $\neg\alpha$, the tableau τ is finished with root $\neg\alpha$ and non-contradictory. Let π be any finished non-contradictory path through τ (there is at least one). Then the propositions on π form a Hintikka set, so are satisfiable. Since $\neg\alpha$ is on π, it is satisfiable as well. Thus, α is not a tautology. □

Remark 12.2.3 The method of truth tables provide one way to decide for any proposition α whether or not α is a tautology We now have a second procedure using the method of tableaux. To check is α is a tautology, construct a finished tableau for $\mathbf{F}\alpha$ using one of the procedure that guarantee the finished tableau is finite. A procedure such as that in the proof of theorem **??**, or the CST construction, or the method in exercise **??**. If there is a non-contradictory path, you can construct a valuation from the signed propositional symbols in which α is false; otherwise, the tableau is contradictory and α is a tautology.

The method of tableaux forms the rationale of the logic programming language `PROLOG` and many other constructive theorem provers as programming languages. One starts with an assumption such as "thre is no x such that $\mathcal{P}(x)$" where $\mathcal{P}$ is some condition expressible in the language and one either proves it true or finds a counterexample, that is, one actually produces an x for which $\mathcal{P}(x)$. The conditions $\mathcal{P}$ describable in the language are expressible in first-order, not propositional logic, but we will extend the method of tableaux to first-order logic and prove similar a soundness and completeness theorem for this extension of tableaux as well. There is an important dichotomy though between the propositional and first-order constructions of tableaux: there are valid first-order formulae whose finished tableaux is infinite. However, we will show that if a first-order formula is valid (it cannot be false), then we can construct a contradictory finite tableau.