Applied Logic	**Handout for Lecture 18**
CS 4860 Spring 2009	**Thursday, March 26, 2009**

Completeness

A *Hintikka Set for a universe U* is a set S of U-formulas such that for all closed U-formulas A, α, β, γ, and δ the following conditions hold.

H_0 : A atomic and $A \in S \mapsto \bar{A} \notin S$
H_1 : $\alpha \in S \mapsto \alpha_1 \in S \wedge \alpha_2 \in S$
H_2 : $\beta \in S \mapsto \beta_1 \in S \vee \beta_2 \in S$
H_3 : $\gamma \in S \mapsto \forall k \in U.\ \gamma(k) \in S$
H_4 : $\delta \in S \mapsto \exists k \in U.\ \delta(k) \in S$

Hintikka Lemma: $\forall U \neq \varnothing.\ \forall S\text{:Set(Form}_U\text{)}.\ (\text{Hintikka}(S) \mapsto \exists I\text{:Pred}_S \to \text{Rel(U)}.\ U,I \models S)$

Proof: *Because of axiom H_0 we know how to define an intepretation that satisfies all the atomic formulas in S.*

Define $I(P(k_1, ..., k_n)) = \begin{cases} \texttt{f} & \text{if } FP(k_1, ..., k_n) \in S \\ \texttt{t} & \text{otherwise} \end{cases}$

Then I maps all the predicate symbols in S to relations over U. What remains to be shown is $\forall Y \in S.\ U,I \models Y$. We prove this by structural induction on formulas, keeping in mind that the cases for γ and δ are straightforward generalizations of those for α and β.

base case: If Y is an atomic formula then by definition $Y \in S \mapsto I(Y)=t \mapsto U,I \models Y$.

step case: Assume the the claim holds for all subformulas of Y.

- If Y is of type α then $\alpha_1, \alpha_2 \in S$, hence $U,I \models \alpha_1$ and $U,I \models \alpha_2$. By definition of first-order valuations $U,I \models Y$.
- If Y is of type β then $\beta_1 \in S$ or $\beta_2 \in S$, hence $U,I \models \beta_1$ or $U,I \models \beta_2$ and thus $U,I \models Y$.
- If Y is of type γ then $\gamma(k) \in S$ for all $k \in U$, hence by induction $U,I \models \gamma(k)$ for all k and by definition of first-order valuations $U,I \models Y$.
- If Y is of type δ then $\delta(k) \in S$ for some $k \in U$, hence by induction $U,I \models \delta(k)$ for some k and by definition of first-order valuations $U,I \models Y$.

A systematic procedure for proving a first-order formula X:

Start with the signed formula FX and recursively extend the tableau as follows:

- If the tableau is already closed then stop. The formula is valid.
- Otherwise select a node Y in the tableau that is of *minimal level* wrt. the still unused nodes and extend *every* open branch θ through Y as follows:
  - If Y is α extend θ to $\theta \cup \{\alpha_1, \alpha_2\}$.
  - If Y is β, extend θ to two branches $\theta \cup \{\beta_1\}$ and $\theta \cup \{\beta_2\}$.
  - If Y is γ, extend θ to $\theta \cup \{\gamma(a), \gamma\}$, where a is the first parameter that does not yet occur on θ.
  - If Y is δ, extend θ to $\theta \cup \{\delta(a)\}$, where a is the first parameter that does not yet occur in the tableau tree.

A systematic tableau is *finished*, if it is either infinite or finite and cannot be extended any further.

Lemma: In every finished systematic tableau, every open branch is a Hintikka sequence.

Corollary: In every finished systematic tableau, every open branch is uniformly satisfiable.

Theorem (Completeness theorem for first-order logic):

If a first-order formula X is valid, then X is provable. Furthermore the systematic tableau method will construct a closed tableau for FX after finitely many steps.

Corollary: If a first-order formula X is valid, then X the there is an atomically closed tableau for FX.

Theorem (Löwenheim theorem for first-order logic):

If a first-order formula X is satisfiable, then it is satisfiable in a denumerable domain.

Compactness

A *first-order tableau for a set S of pure formulas* starts with an arbitrary element of S at its origin and is then extended by applying one of the 4 rules α, β, γ, or δ, or by adding another element of S to the end of an open branch. The elements of S so added are the *premises* of the tableau. We call a tableau *complete* if every open branch is a Hintikka set for the universe of parameters and contains all the elements of S.

Lemma: For every denumerable set S there is a complete tableau for S.

Proof: We construct the desired tableau by combining our systematic proof procedure with the construction of a tableau for S that we used in the propositional case.

Arrange S as a denumerable sequence $X_1, X_2, \ldots, X_n, \ldots$. In stage 1 place X_1 at the origin of the tableau. In stage $n+1$ extend the tableau constructed at stage n as follows.

- If the tableau is already closed then stop. The formula is valid.
- Otherwise select a node Y in the tableau that is of *minimal level* wrt. the still unused nodes and extend *every* open branch θ through Y as in the systematic procedure and add X_{n+1} to the end of every open branch.

By construction every open branch in the resulting tableau is a Hintikka set for the universe of parameters (we used the systematic method) and contains the set S.

Lemma: If a pure set S has a closed tableau, then a finite subset of S is unsatisfiable.

Proof: Assume S has a closed tableau $\mathcal{T}$ and consider the set S_p of premises of $\mathcal{T}$. By König's lemma, $\mathcal{T}$ must be finite and so is S_p. S_p must be unsatisfiable, since otherwise every branch containing S_p would be open.

Theorem: If all finite subsets of a denumerable set S of pure formulas are satisfiable, then S is uniformly satisfiable in a denumerable domain.

Proof: Let $\mathcal{T}$ be a complete tableau for S. Since all finite subsets of S are satisfiable, $\mathcal{T}$ cannot be closed, so it has an open branch θ. Since $\mathcal{T}$ is complete, θ is a Hintikka for the denumerable universe of parameters that contains S. Thus S is uniformly satisfiable in a denumerable universe.

Corollary: (**Compactness of First-Order Logic**)

If all finite subsets of a pure set S are satisfiable, then S is uniformly satisfiable

Corollary: (**Skolem-Löwenheim theorem for First-Order Logic**)

If a pure set S of is satisfiable then it is satisfiable in a denumerable domain.

Corollary: If no tableau for a pure set S can close, then S is satisfiable in a denumerable domain.