Herbrand Models

Let $\mathcal{L}$ be a first-order language with set of function symbols $\mathcal{F}$. The set $\mathcal{G}$ of all ground (i.e., variable-free) terms built from function symbols in $\mathcal{F}$ is called the *Herbrand universe* (over $\mathcal{F}$).

Note that the Herbrand universe is the empty set if $\mathcal{F}$ contains no constant. We shall consider only sets $\mathcal{F}$ that contain at least one constant.

For each function symbol $f \in \mathcal{F}$ we denote by $f^{\mathcal{G}}$ the mapping from $\mathcal{G}^n$ to $\mathcal{G}$ defined by:

$$f^{\mathcal{G}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n),$$

for all terms $t_1, \ldots, t_n$ in $\mathcal{G}$.

A model $\mathcal{M}$ for the language $\mathcal{L}$ is said to be a *Herbrand model* if

1. its universe is the Herbrand universe $\mathcal{G}$ and
2. $f^{\mathcal{M}}$ is the function $f^{\mathcal{G}}$, for all function symbols f.

A sentence ϕ is said to be *Herbrand satisfiable* if there is a Herbrand model $\mathcal{M}$ such that $\mathcal{M} \models \phi$.

Theorem

A universal sentence ϕ is satisfiable if, and only if, it is Herbrand satisfiable.

Literals and Clauses

A *literal* is either an atomic formula, $P(t_1, \ldots, t_n)$, (a *positive* literal) or the negation thereof, $\neg P(t_1, \ldots, t_n)$, (a *negative* literal).

A *clause* is a multiset of literals,

$$[L_1, \ldots, L_n],$$

usually written as a (generalized) disjunction,

$$L_1 \vee \cdots \vee L_n.$$

If $n = 1$ we speak of a *unit clause*; if $n = 0$ of the *empty clause*.

Literals and clauses without variables are said to be *ground*.

For example, $P(a, b) \vee \neg Q(f(a)) \vee Q(f(b))$ is a ground clause and $P(x, f(x))$ is a unit clause containing the variable x.

If C is a clause $L_1 \vee \cdots \vee L_n$, then any sentence

$$\forall x_1 \ldots \forall x_k \, (L_1 \vee \cdots \vee L_n)$$

where $x_1, \ldots, x_k$ are all the variables occurring in C, is called a *universal closure* of C (and, ambiguously, denoted by ϕ_C).

For example, $\forall x \forall y [\neg P(x) \vee Q(y)]$ is a universal closure of $\neg P(x) \vee Q(y)$.

Semantics of Clauses

A ground clause $C = L_1 \vee \cdots \vee L_n$ is said to be *true* in a Herbrand model $\mathcal{M}$, written $\mathcal{M} \models C$, if at least one of its literals is true in $\mathcal{M}$; and *false*, otherwise. (The definition reflects the fact that clauses represent disjunctions.)

Note that the empty clause is false in all Herbrand models.

The semantics of clauses with variables is defined via instantiation.

Let $\mathcal{V}$ be the set of variables of the given language $\mathcal{L}$ and let $\sigma : \mathcal{V} \rightarrow \mathcal{T}$ be a substitution from variables to terms.

If C is a clause $L_1 \vee \cdots \vee L_n$, we denote by $C\sigma$ the clause $L_1\sigma \vee \cdots \vee L_n\sigma$ obtained from C by applying substitution σ.

We say that $C\sigma$ is an *instance* of C and speak of a *ground instance* if $C\sigma$ contains no variables.

For example, $P(f(x))$ and $P(f(a))$ are instances of $P(x)$.

A clause C is said to be *true* in a Herbrand model $\mathcal{M}$, written $\mathcal{M} \models C$, if all of its ground instances are true in $\mathcal{M}$; and *false*, otherwise.

Herbrand Satisfiability of Clauses

A clause C is said to be *Herbrand satisfiable* if it is true in some Herbrand model; and *Herbrand unsatisfiable* otherwise.

A set of clauses S is said to be *Herbrand satisfiable* if there exists a Herbrand model $\mathcal{M}$ in which all clauses $C \in S$ are true. Otherwise, S is said to be *Herbrand unsatisfiable*.

Proposition.

> A set of clauses S is Herbrand satisfiable if, and only if, the set of all ground instances of clauses in S is Herbrand satisfiable.

Since the Herbrand universe is usually infinite, a clause with variables may have infinitely many ground instances. (In other words, function symbols and variables provide a finite description mechanism for infinite sets of clauses.)

Satisfiability of Clauses

A set S of clauses is said to be *satisfiable* if there exists a model $\mathcal{M}$ such that

$$\mathcal{M} \models \phi_C$$

for all clauses C in S.

Theorem

1. A set of clauses is satisfiable if, and only if, it is Herbrand satisfiable.

2. For every predicate logic sentence ϕ one can effectively construct a finite set of clauses S such that ϕ is satisfiable if, and only if, S is (Herbrand) satisfiable.

Sketch of proof. For the second part we may use Skolemization and other techniques discussed previously to construct a universal sentence

$$\forall x_1 \ldots \forall x_n \, \psi$$

that is satisfiable if, and only if, ϕ is satisfiable. The quantifier-free formula ψ can be transformed to conjunctive form,

$$\psi_1 \wedge \cdots \wedge \psi_k,$$

where each formula ψ_i is a generalized disjunction of literals,

$$L_{i1} \vee \cdots \vee L_{ik_i}.$$

The corresponding (finite) set of clauses is satisfiable if, and only if ϕ is satisfiable.

Semantic Trees

Semantic trees provide a way of systematically exploring Herbrand models for a given first-order language.

Let $A_1, A_2, \ldots$ be a listing of all the ground atomic formulas (other than $\top$ and $\bot$) in the given language. The corresponding semantic tree is a labeled binary tree, inductively defined by:

1. The root of the tree is labeled by $\top$.

2. Each node at depth $k \geq 0$ has two children labeled by A_k and $\neg A_k$, respectively.

A semantic tree may be finite or infinite, depending on the number of ground atomic formulas.

For example, if a language contains only two predicate constants, P and Q, then each semantic tree is a complete binary tree of height 2.

Note that the literals A_k and $\neg A_k$ occur only at depth k in the semantic tree.

The key observation is that

> *the branches of a semantic tree represent all possible Herbrand models for the given language.*

Representation of Herbrand Models

Herbrand models can be represented by sets of literals (or also by sets of atomic formulas).

More specifically, if I is a set of literals containing no complementary pairs of literals, we obtain a corresponding Herbrand model $\mathcal{M}_I$ as follows:

> If P is a predicate constant, then $P^{\mathcal{M}_I}$ is true if, and only if, $P \in I$.
>
> If P is of arity $n > 0$, then
>
> $$P^{\mathcal{M}_I} = \{(t_1, \ldots, t_n) : P(t_1, \ldots, t_n) \in I\}.$$

For simplicity we often speak of an "model I," or of a formula being "true in I," etc. instead of referring more precisely to $\mathcal{M}_I$.

A branch (i.e., a maximal path from the root) of a semantic tree represents a Herbrand model corresponding to the set of all literals labeling its nodes.

Lemma

> Each Herbrand model for a given first-order language is represented by a branch in a corresponding semantic tree.

Failure Nodes

Let N be a set of ground clauses and T be a semantic tree. We can classify nodes in T with respect to N as follows.

If p is a node in T, let I_p denote the set of all literals (labeling nodes) on the path from the root to p.

We call p a *failure node* (with respect to N) if there is a clause C in N such that I_p contains the complement of each literal in C. If N contains no such clause, then p is a called a *non-failure node* (for N).

Lemma.

(a) If p is a failure node with respect to N, so are all its descendants.

(b) If a branch of T contains no failure nodes for N, then the set of all its literals describes a Herbrand model in which all formulas of N are true.

The lemma implies that if a set N is unsatisfiable, then each branch of T must contain a failure node.

Refutation Trees

Let N be a set of ground clauses.

A *minimal refutation tree* is obtained from a semantic tree for N by deleting all nodes that have a failure node as ancestor.

Note that all failure nodes in a minimal refutation tree must be leaves.

Given N and T, the corresponding minimal refutation tree is unique.

Lemma.

> If N is unsatisfiable, then each minimal refutation tree for N is finite.

The lemma can be proved by König's Infinity Lemma.

The Infinity Lemma

Wehave already encountered the Infinity Lemma when discussing the Davis-Putnam method.

König's Infinity Lemma

> A finitely branching, infinite tree has an infinite branch.

For other applications of the Infinity Lemma, see the discussion in D.E. Knuth, *The Art of Computer Programming*, vol. 1, pp. 382–386.