Logic Programs and Minimal Models

Marco Piastra
Logic Program

- An example of logic program:

  \[ \Pi \equiv \{ \{ \text{Human}(x), \neg \text{Philosopher}(x) \} , \{ \text{Mortal}(y), \neg \text{Human}(y) \} , \{ \text{Philosopher}(\text{socrates}) \} , \{ \text{Philosopher}(\text{plato}) \} , \{ \text{Philosopher}(\text{aristotle}) \} \} \]

  \[ \phi \equiv \exists x \text{ Mortal}(x) \]
  \[ \neg \phi \equiv \neg \exists x \text{ Mortal}(x) \]
  \[ \equiv \forall x \neg \text{ Mortal}(x) \]
  \[ \equiv \{ \neg \text{ Mortal}(x) \} \] (a goal, i.e. a Horn clause)

By applying resolution in an exhaustive way, we obtain:

  \[ \Sigma \equiv \{ [x/\text{socrates}] , [x/\text{plato}] , [x/\text{aristotle}] \} \]

Looks like a query on an implicit database ...

- Answer Set

  It includes all complete substitutions of the variables in the goal corresponding to the closed branches (i.e. with an empty clause) in the SLD tree
Herbrand Universe, Herbrand Base

- **Herbrand terms and atoms**
  Given a signature $\Sigma$
  
  A Herbrand term is a *ground term* (i.e. a term that contains no variables)
  
  Examples:
  
  \[ f(a), g(a,b), g(f(a),b), g(f(a),g(b,c)), g(f(a),g(f(b),c)), \ldots \]

  A Herbrand atom is a *ground atom* (i.e. an atom that contains no variables)
  
  Examples:
  
  \[ P(f(a)), P(g(a,b)), Q(g(f(a),b), g(f(a),g(b,c))), \ldots \]

- **Herbrand universe**
  
  The set of all Herbrand terms from $\Sigma$
  
  Example:
  
  \[ U_H \equiv \{ f(a), g(a,b), g(f(a),b), g(f(a),g(b,c)), g(f(a),g(f(b),c)), \ldots \} \]

- **Herbrand base**
  
  The set of all Herbrand atoms from $\Sigma$
  
  Example:
  
  \[ B_H \equiv \{ P(f(a)), P(g(a,b)), Q(g(f(a),b), g(f(a),g(b,c))), \ldots \} \]
Herbrand models

- **Herbrand structure**
  A semantic structure \(<U_H, \Sigma, v_H>\) such that

- **Herbrand interpretation** \(v_H\)
  For constants, \(v_H(c) = c\)
  For ground terms, \(v_H(t) = t\)
  For predicate symbols, \(v_H \subseteq B_H\)
    i.e. a **subset** of the Herbrand base \(B_H\)
    Example: \(v_H \equiv \{ P(a), P(f(b)), P(c), Q(a,g(b,c)), Q(b,c) \ldots \}\)

- **Herbrand model**
  \(\varphi \in \text{Atom}(L_{PO}), \quad \langle U_H, \Sigma, v_H \rangle[s] \models \varphi \quad \text{iff} \quad \varphi \in v_H\)
  \(\varphi \in \text{Atom}(L_{PO}), \quad \langle U_H, \Sigma, v_H \rangle[s] \models \neg \varphi \quad \text{iff} \quad \varphi \notin v_H\)
  \(\langle U_H, \Sigma, v_H \rangle[s] \models \neg \varphi \quad \text{iff} \quad \langle U_H, \Sigma, v_H \rangle[s] \not\models \varphi\)
  \(\langle U_H, \Sigma, v_H \rangle[s] \models \varphi \rightarrow \psi \quad \text{iff} \quad (\langle U_H, \Sigma, v_H \rangle[s] \not\models \varphi \text{ or } \langle U_H, \Sigma, v_H \rangle[s] \models \psi)\)
  \(\langle U_H, \Sigma, v_H \rangle[s] \models \forall x \varphi \quad \text{iff for all } \ c \in \text{Cost}(L_{PO}), \quad \langle U_H, \Sigma, v_H \rangle[s](x:c) \models \varphi\)
Horn clauses and Herbrand models

- **Herbrand Theorem**
  Given a theory of universal sentences $\Phi$, $H(\Phi)$ has a model iff $\Phi$ has a model

- **Corollary (for Horn clauses)**
  Given a set $\Phi$ of Horn clauses, the two following statements are equivalent:
  - $\Phi$ is satisfiable
  - $\Phi$ has an Herbrand model
  This is not true in general: only if $\Phi$ is a set of Horn clauses
Horn Clauses and Herbrand Models

- **Corollary to Herbrand theorem (for Horn clauses)**
  
  Given a set $\Phi$ of Horn clauses, the two following statements are equivalent:
  - $\Phi$ is satisfiable
  - $\Phi$ has an Herbrand model

  This is not true in general: only if $\Phi$ is a set of Horn clauses

- **Herbrand minimal model**

  The minimal model $M_{\Phi}$ for a set of Horn clauses $\Phi$ is:
  
  $$ M_{\Phi} \equiv \bigcap_{i} M_{\phi_i} $$

  where $M_{\phi_i}$ is a Herbrand model of $\Phi$

- **Theorem** (van Emde Boas Kowalski, 1976)

  Let $\Phi$ be a set of Horn clauses and $\varphi$ a ground atom.

  These three statements are equivalent:
  - $\Phi \models \varphi$
  - $\varphi \in M_{\Phi}$
  - $\varphi$ is derivable from $\Phi$ via resolution with refutation
Logic programming system and minimal model

- **Theorem** (Apte et al., 1982)
  Let $\Pi$ be a *logical program* (i.e. a set of definite clauses).
  The (finite) success set of $\Pi$ with SLD-resolution *(fair)* coincides with $M_\Pi$

- A logic programming system (i.e. Prolog) can generate the *subset* of $M_\Pi$ corresponding to a specific **goal**
  A goal $\{ \neg \alpha_1, \neg \alpha_2, ..., \neg \alpha_m \}$ where the variables $x_1, x_2, ..., x_m$ occur
  is equivalent to the sentence $\forall x_1 \forall x_2 ... \forall x_n (\neg \alpha_1 \lor \neg \alpha_2 \lor ... \lor \neg \alpha_m)$
  which is equivalent to $\neg \exists x_1 \exists x_2 ... \exists x_n (\alpha_1 \land \alpha_2 \land ... \land \alpha_m)$
  A logic programming system can generate all possible **substitutions** $[x_1/t_1, x_2/t_2, ..., x_n/t_n]$ such that $\Pi \cup \{ \neg (\alpha_1 \land \alpha_2 \land ... \land \alpha_m)[x_1/t_1, x_2/t_2, ..., x_n/t_n]\}$ is unsatisfiable
    (that implies $\Pi \models (\alpha_1 \land \alpha_2 \land ... \land \alpha_m)[x_1/t_1, x_2/t_2, ..., x_n/t_n]$)
    (that implies $(\alpha_1 \land \alpha_2 \land ... \land \alpha_m)[x_1/t_1, x_2/t_2, ..., x_n/t_n] \in M_\Pi$)
  Each goal act like a *filter*, i.e. defining the subset of $M_\Pi$

*NOTE: a logic programming system with a **fair** strategy can do so...*