Definite Logic Programs: Models

CSE 505 – Computing with Logic
Stony Brook University
htTp://www.cs.stonybrook.edu/~cse505
Logical Consequences of Formulae

- Recall: F is a logical consequence of P (i.e. $P \models F$) iff
  Every model of P is also a model of F.
- Since there are (in general) infinitely many possible interpretations, how can we check if F is a logical consequence of P?
  - Solution: choose one "canonical" model I such that
    $I \models P$ and $I \models F \implies P \models F$
Definite Clauses

- A formula of the form $p(t_1, t_2, \ldots, t_n)$, where $p/n$ is an $n$-ary predicate symbol and $t_i$ are all terms is said to be atomic.

- If $A$ is an atomic formula then
  - $A$ is said to be a positive literal
  - $\neg A$ is said to be a negative literal

- A formula of the form $\forall (L_1 \lor L_2 \lor \ldots \lor L_n)$ where each $L_i$ is a literal (negative or positive) is called a clause.

- A clause $\forall (L_1 \lor L_2 \lor \ldots \lor L_n)$ where exactly one literal is positive is called a definite clause (also called Horn clause).

- A definite clause is usually written as:
  - $\forall (A_0 \lor \neg A_1 \lor \ldots \lor \neg A_n)$
  - or equivalently as $A_0 \leftarrow A_1, A_2, \ldots, A_n$.

- A definite program is a set of definite clauses.
Herbrand Universe

• Given an alphabet A, the set of all **ground terms** constructed from the constant and function symbols of A is called the **Herbrand Universe** of A (denoted by $U_A$).

• Consider the program:

  \[
  \begin{align*}
  p(zero). \\
  p(s(s(X))) & \leftarrow p(X). 
  \end{align*}
  \]

• The Herbrand Universe of the program's alphabet is: $U_A = \{\text{zero}, s(\text{zero}), s(s(\text{zero})) , \ldots\}$
Herbrand Universe: Example

• Consider the "relations" program:

```
parent(pam, bob).    parent(bob, ann).
parent(tom, bob).    parent(bob, pat).
parent(tom, liz).    parent(pat, jim).
grandparent(X,Y) :-
    parent(X,Z), parent(Z,Y).
```

• The Herbrand Universe of the program's alphabet is:

\[ U_A = \{ \text{pam, bob, tom, liz, ann, pat, jim} \} \]
Given an alphabet A, the set of all ground atomic formulas over A is called the \textit{Herbrand Base} of A (denoted by $B_A$).

Consider the program:

\[
\begin{align*}
p(zero) \\
p(s(s(X))) & \leftarrow \ p(X).
\end{align*}
\]

The Herbrand Base of the program's alphabet is: $B_A = \{p(zero), \ p(s(zero)), \ p(s(s(zero))), \ldots \}$
Herbrand Base: Example

- Consider the "relations" program:

  
  \[
  \begin{align*}
  \text{parent}(pam, \text{bob}). & \quad \text{parent}(\text{bob}, \text{ann}). \\
  \text{parent}(\text{tom}, \text{bob}). & \quad \text{parent}(\text{bob}, \text{pat}). \\
  \text{parent}(\text{tom}, \text{liz}). & \quad \text{parent}(\text{pat}, \text{jim}). \\
  \text{grandparent}(X,Y) : - & \\
  \quad \text{parent}(X,Z), \text{parent}(Z,Y). \\
  \end{align*}
  \]

- The Herbrand Base of the program's alphabet is:

  \[
  B_A = \{ \text{parent}(pam, pam), \text{parent}(pam, bob), \text{parent}(pam, tom), ..., \text{parent}(bob, pam), ..., \text{grandparent}(pam,pam), ..., \text{grandparent}(bob,pam), ... \}.
  \]
Herbrand Interpretations and Models

- A **Herbrand Interpretation** of a program P is an interpretation I such that:
  - The domain of the interpretation: $|I| = U_P$
  - For every constant $c$: $c_I = c$
  - For every function symbol $f/n$:
    $f_I(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$
  - For every predicate symbol $p/n$:
    $p_I \subseteq (U_P)^n$ (i.e. some subset of $n$-tuples of ground terms)

- A **Herbrand Model** of a program P is a Herbrand interpretation that is a model of P.
Herbrand Models

- All Herbrand interpretations of a program give the same “meaning” to the constant and function symbols.
- Different Herbrand interpretations differ only in the “meaning” they give to the predicate symbols.
- We often write a Herbrand model simply by listing the subset of the Herbrand base that is true in the model.
- Example: Consider our numbers program, where \{p(zero), p(s(s(zero))), p(s(s(s(s(zero)))))), ...\} represents the Herbrand model that treats \p_{I}={\text{zero}, s(s(\text{zero})), s(s(s(s(\text{zero}))))}, \ldots\} as the meaning of \(p\).
Properties of Herbrand Models

1) If $M$ is a set of Herbrand Models of a definite program $P$, then $\cap M$ is also a Herbrand Model of $P$.

2) For every definite program $P$ there is a unique least model $M_p$ such that:
   - $M_p$ is a Herbrand Model of $P$ and,
   - for every Herbrand Model $M$, $M_p \subseteq M$.

3) For any definite program, if every Herbrand Model of $P$ is also a Herbrand Model of $F$, then $P \models F$.

4) $M_p = \text{the set of all ground logical consequences of } P$. 
Sufficiency of Herbrand Models

- Let $P$ be a definite program. If $I'$ is a model of $P$ then $I = \{A \in B_P \mid I' \models A\}$ is a Herbrand model of $P$.

Proof (by contradiction):

Let $I$ be a Herbrand interpretation.

Assume that $I'$ is a model of $P$ but $I$ is not a model.

Then there is some ground instance of a clause in $P$:

$$A_0 :\neg A_1, \ldots, A_n.$$  

which is not true in $I$ i.e., $I \models A_1, \ldots, I \models A_n$ but $I \not\models A_0$.

By definition of $I$ then, $I' \models A_1, \ldots, I' \models A_n$ but $I' \not\models A_0$.

Thus, $I'$ is not a model of $P$, which contradicts our earlier assumption.
Definite programs only

- Let $P$ be a definite program. If $I'$ is a model of $P$ then $I = \{ A \in B_p \mid I' \models A \}$ is a Herbrand model of $P$.
  - This holds only for definite programs.
- Consider $P = \{ \neg p(a), \exists X.p(X) \}$
  - There are two Herbrand interpretations: $I_1 = \{ p(a) \}$ and $I_2 = \{ \}$
  - The first is not a model of $P$ since $I_1 \not\models \neg p(a)$.
  - The second is not a model of $P$ since $I_2 \not\models \exists X.p(X)$
  - But there is a non-Herbrand model $I$:
    - $| I | = \mathbb{N}$, the set of natural numbers
    - $a_I = 0$
    - $p_I = \text{“is odd”}$
Properties of Herbrand Models

• If $M_1$ and $M_2$ are Herbrand models of $P$, then $M = M_1 \cap M_2$ is a model of $P$.

• Assume $M$ is not a model.

• Then there is some clause $A_0 : \neg A_1, \ldots, A_n$ such that $M \models A_1, \ldots, M \models A_n$ but $M \not\models A_0$.

• Which means $A_0 \notin M_1$ or $A_0 \notin M_2$.

• But $A_1, \ldots, A_n \in M_1$ as well as $M_2$.

• Hence one of $M_1$ or $M_2$ is not a model.
Properties of Herbrand Models

• There is a unique least Herbrand model
• Let $M_1$ and $M_2$ are two incomparable minimal Herbrand models, i.e.,
  $M = M_1 \cap M_2$ is also a Herbrand model (previous theorem), and $M \subseteq M_1$ and $M \subseteq M_2$
• Thus $M_1$ and $M_2$ are not minimal
Least Herbrand Model

• The least Herbrand model $M_p$ of a definite program $P$ is the set of all ground logical consequences of the program.

$$M_p = \{ A \in B_p \mid P \models A \}$$

• First, $M_p \supseteq \{ A \in B_p \mid P \models A \}$:
  • By definition of logical consequence, $P \models A$ means that $A$ has to be in every model of $P$ and hence also in the least Herbrand model.
Least Herbrand Model

- Second, $Mp \subseteq \{A \in Bp \mid P \models A\}$:
  - If $Mp \models A$ then $A$ is in every Herbrand model of $P$.
  - But assume there is some model $I' \models \neg A$.
  - By sufficiency of Herbrand models, there is some Herbrand model $I$ such that $I \models \neg A$.
  - Hence $A$ is not in some Herbrand model, and hence is not in $Mp$. 
Finding the Least Herbrand Model

• Immediate consequence operator:
  • Given $I \subseteq B_p$, construct $I'$ such that
    $$I' = \{ A_0 \in B_p \mid A_0 \leftarrow A_1, \ldots, A_n \text{ is a ground instance of a clause in } P \text{ and } A_1, \ldots, A_n \in I \}$$
  • $I'$ is said to be the immediate consequence of $I$.
  • Written as $I' = T_p(I)$, $T_p$ is called the *immediate consequence* operator.

• Consider the sequence:
  $$\emptyset, T_p(\emptyset), T_p(T_p(\emptyset)), \ldots, T_p^i(\emptyset), \ldots$$
  • $M_p \supseteq T_p^i(\emptyset)$ for all $i$.
  • Let $T_p \uparrow \omega = \bigcup_{i=0,\infty} T_p^i(\emptyset)$
  • Then $M_p \subseteq T_p \uparrow \omega$
Computing Least Herbrand Models: An Example

\begin{align*}
\text{parent}(pam, \text{bob}). \\
\text{parent}(tom, \text{bob}). \\
\text{parent}(tom, \text{liz}). \\
\text{parent}(bob, \text{ann}). \\
\text{parent}(bob, \text{pat}). \\
\text{parent}(pat, \text{jim}). \\
\text{anc}(X,Y) :&= \text{parent}(X,Y). \\
\text{anc}(X,Y) :&= \text{parent}(X,Z), \text{anc}(Z,Y).
\end{align*}

\begin{align*}
M_1 &= \emptyset \\
M_2 &= \mathcal{T}_P(M_1) = \{ \text{parent}(pam,\text{bob}), \\
&\quad \text{parent}(tom,\text{bob}), \\
&\quad \text{parent}(tom,\text{liz}), \\
&\quad \text{parent}(bob,\text{ann}), \\
&\quad \text{parent}(bob,\text{pat}), \\
&\quad \text{parent}(pat,\text{jim}) \} \\
M_3 &= \mathcal{T}_P(M_2) = \{ \text{anc}(pam,\text{bob}), \text{anc}(tom,\text{bob}), \\
&\quad \text{anc}(tom,\text{liz}), \text{anc}(bob,\text{ann}), \\
&\quad \text{anc}(bob,\text{pat}), \text{anc}(pat,\text{jim}) \} \cup M_2 \\
M_4 &= \mathcal{T}_P(M_3) = \{ \text{anc}(pam,\text{ann}), \text{anc}(pam,\text{pat}), \\
&\quad \text{anc}(tom,\text{ann}), \text{anc}(tom,\text{pat}), \\
&\quad \text{anc}(bob,\text{jim}) \} \cup M_3 \\
M_5 &= \mathcal{T}_P(M_4) = \{ \text{anc}(pam,\text{jim}), \{ \text{anc}(tom,\text{jim}) \} \\
&\quad \cup M_4 \\
M_6 &= \mathcal{T}_P(M_5) = M_5
\end{align*}
Computing Mp: Practical Considerations

- Computing the least Herbrand model, Mp, as the least fixed point of Tp:
  - terminates for Datalog programs (programs w/o function symbols)
  - may not terminate in general.
- For programs with function symbols, computing logical consequence by first computing Mp is impractical.
- Even for Datalog programs, computing least fixed point directly using the Tp operator is wasteful (known as Naive evaluation).
- Note that $T_p^i(\emptyset) \subseteq T_p^{i+1}(\emptyset)$.
- We can calculate $\Delta T_p^{i+1}(\emptyset) = T_p^{i+1}(\emptyset) - T_p^i(\emptyset)$ [The difference between the sets computed in two successive iterations] This strategy is known as semi-naive evaluation.