Logic Programming

Theory Lecture 6: Fixed Points and Herbrand Models

Richard Mayr

School of Informatics

30. October 2014

Recap (Lecture 3): Definite clause predicate logic

A *definite clause* is a formula of one of the two shapes below

$$B \qquad \text{(a Prolog } \textit{fact } B\,.\text{)}$$

$$A_1 \wedge \cdots \wedge A_k \;\to B \qquad \text{(a Prolog } \textit{rule } B \;\text{:-}\; A_1, \ldots, A_k.\text{)}$$

where $A_1, \ldots, A_k, B$ are all *atomic formulas*.

A *logic program* is a list $F_1, \ldots, F_n$ of definite clauses

A *goal* is a list $G_1, \ldots, G_m$ of atomic formulas.

The job of the system is to ascertain whether the logical consequence below holds.

$$\forall Vars(F_1).\, F_1, \ldots, \forall Vars(F_n).\, F_n \;\models\; \exists Vars(G_1, \ldots, G_m).\, G_1 \wedge \cdots \wedge G_m$$

Recap (Lecture 5): The minimum Herbrand model

We define the structure $\mathcal{H}$ as follows.

- The universe is the *Herbrand universe*: the set of all ground terms.
- A constant $\mathtt{c}$ is interpreted by $\mathtt{c}^{\mathcal{H}} = \mathtt{c}$.
- A function symbol $\mathtt{f}/k$ is interpreted by $\mathtt{f}^{\mathcal{H}}(u_1, \ldots, u_k) = \mathtt{f}(u_1, \ldots, u_k)$.
- A predicate symbol $\mathtt{p}/k$ is interpreted by

$$\mathtt{p}^{\mathcal{H}}(u_1, \ldots, u_k) = \textbf{true} \;\Leftrightarrow\; \text{the goal } \mathtt{p}(u_1, \ldots, u_k) \text{ is derivable}$$

The minimum Herbrand model $\mathcal{H}$ is indeed a model of the program $F_1, \ldots, F_n$, i.e., for every F_i, we have $\mathcal{H} \models \forall \mathit{Vars}(F_i).\, F_i$.

Importance of minimum Herbrand model

Theorem

The logical consequence

$$\forall Vars(F_1).\, F_1, \ldots, \forall Vars(F_n).\, F_n \models \exists Vars(G_1, \ldots, G_m).\, G_1 \wedge \cdots \wedge G_m$$

holds if and only if

$$\mathcal{H} \models \exists Vars(G_1, \ldots, G_m).\, G_1 \wedge \cdots \wedge G_m$$

In other words, an (implicitly existentially quantified) goal $G_1, \ldots, G_m$ is a logical consequence of a program if and only if it is true in the minimal Herbrand model of the program.

Thus we can understand Prolog programs and queries as being tools for exploring truth in this special model.

Proof of Theorem

If the logical consequence

$$\forall Vars(F_1).\, F_1, \ldots, \forall Vars(F_n).\, F_n \models \exists Vars(G_1, \ldots, G_m).\, G_1 \land \cdots \land G_m$$

holds then

$$\mathcal{H} \models \exists Vars(G_1, \ldots, G_m).\, G_1 \land \cdots \land G_m$$

because $\mathcal{H}$ is a model of the program.

Conversely, if

$$\mathcal{H} \models \exists Vars(G_1, \ldots, G_m).\, G_1 \land \cdots \land G_m$$

then, by Lecture 5 Proof of completeness (completed), the goal $G_1, \ldots, G_m$ is derivable by SLD resolution. Whence, by Lecture 5 soundness of inference system:

$$\forall Vars(F_1).\, F_1, \ldots, \forall Vars(F_n).\, F_n \models \exists Vars(G_1, \ldots, G_m).\, G_1 \land \cdots \land G_m$$

Aim of lecture

The aim of today's lecture is to achieve a better understanding of the minimum Herbrand model.

This will be done using the mathematical notion of *least fixed point*.

To approach this, we first consider some necessary mathematical definitions.

Powersets

A set Y is said to be a *subset* of a set X (notation $Y \subseteq X$) if every member of Y is a member of X, i.e.,

$$\forall z.\ z \in Y \text{ implies } z \in X$$

Given any set X, the set of all subsets of X is called the *powerset* of X, written $\mathcal{P}(X)$.

Example: $\mathcal{P}(\{1,2,3\})$ is a set of eight sets

$$\{\,\{\,\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\,\}$$

Example: $\mathcal{P}(\{\,\})$ is the set $\{\,\{\,\}\,\}$ whose only element is $\{\,\}$
(Often one writes $\emptyset$ for the empty set. So another way of writing the above is $\mathcal{P}(\emptyset) = \{\emptyset\}$.)

Powersets

A set Y is said to be a *subset* of a set X (notation $Y \subseteq X$) if every member of Y is a member of X, i.e.,

$$\forall z.\ z \in Y \text{ implies } z \in X$$

Given any set X, the set of all subsets of X is called the *powerset* of X, written $\mathcal{P}(X)$.

Example: $\mathcal{P}(\{1,2,3\})$ is a set of eight sets

$$\{\,\{\,\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\,\}$$

Example: $\mathcal{P}(\{\,\})$ is the set $\{\,\{\,\}\,\}$ whose only element is $\{\,\}$ (Often one writes $\emptyset$ for the empty set. So another way of writing the above is $\mathcal{P}(\emptyset) = \{\emptyset\}$.)

Note that if X is a finite set with n elements then $\mathcal{P}(X)$ is a finite set with 2^n elements.

Monotone functions

A function $f : \mathcal{P}(X) \to \mathcal{P}(X)$ is said to be *monotone* if, for any pair of subsets $X_1, X_2 \subseteq X$, it holds that

$$X_1 \subseteq X_2 \quad \text{implies} \quad f(X_1) \subseteq f(X_2)$$

Examples Consider $f_1, f_2, f_3 : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$ defined by

$$\begin{aligned} f_1(Y) &= Y \cup \{1\} \\ f_2(Y) &= \begin{cases} \{1\} & \text{if } 1 \in Y \\ \{\,\} & \text{otherwise} \end{cases} \\ f_3(Y) &= \begin{cases} \{\,\} & \text{if } 1 \in Y \\ \{1\} & \text{otherwise} \end{cases} \end{aligned}$$

Then f_1 and f_2 are monotone but f_3 is not.

Fixed points

Given a function $f\colon \mathcal{P}(X) \to \mathcal{P}(X)$.

- A subset $Y \subseteq X$ is said to be a *fixed point* of f if the equation $f(Y) = Y$ holds.
- A subset $Y \subseteq X$ is said to be the *least fixed point* of f if it is a fixed point and, for every fixed point Z of f, it holds that $Y \subseteq Z$.

A function may have zero, one or several fixed points. However, the least fixed point, if it exists, is unique. (Note that "*least*" is a stronger condition than $\subseteq$-minimal.)

Examples Consider $f_1, f_2, f_3\colon \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$ defined on previous slide.

The fixed points of f_1 are $\{1\}, \{1,2\}, \{1,3\}, \{1,2,3\}$. The least fixed point is $\{1\}$.

The fixed points of f_2 are $\{\,\}, \{1\}$. The least fixed point is $\{\,\}$.

The function f_3 has no fixed points.

What does this have to do with logic programming?

We shall view a program P as determining a monotone function

$$f_P \colon \mathcal{P}(X) \to \mathcal{P}(X)$$

where X is the set of *ground atomic formulas*.

The *Herbrand models* of P are in one-to-one correspondence with the fixed points of f_P.

The *minimal Herbrand model* of P corresponds to the least fixed point of f_P.

What does this have to do with logic programming?

We shall view a program P as determining a monotone function

$$f_P \colon \mathcal{P}(X) \to \mathcal{P}(X)$$

where X is the set of *ground atomic formulas*.

The *Herbrand models* of P are in one-to-one correspondence with the fixed points of f_P.

The *minimal Herbrand model* of P corresponds to the least fixed point of f_P.

We first introduce the method by considering the simpler case of propositional Prolog (as considered in Lectures 1 and 2).

Example propositional program

$$
\begin{aligned}
\texttt{arctic} \wedge \texttt{november} &\rightarrow \texttt{noSun} \\
\texttt{australia} \wedge \texttt{november} &\rightarrow \texttt{sun} \\
& \quad\ \texttt{november} \\
& \quad\ \texttt{scotland} \\
\texttt{scotland} &\rightarrow \texttt{arctic}
\end{aligned}
$$

This determines a function f from

$$\mathcal{P}(\{\texttt{arctic}, \texttt{november}, \texttt{noSun}, \texttt{australia}, \texttt{sun}, \texttt{scotland}\})$$

to itself.

$$
\begin{aligned}
f(Y) = \{&\texttt{november}, \texttt{scotland}\} \cup \\
&\{\texttt{noSun} \mid (\texttt{arctic} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
&\{\texttt{sun} \mid (\texttt{australia} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
&\{\texttt{arctic} \mid \texttt{scotland} \in Y\} \cup Y
\end{aligned}
$$

The idea is that $f(Y)$ contains all the atomic facts in the program (in this case `november` and `scotland`) together with all atoms that can be derived from atoms in Y using a single rule of inference in the inference system of Lecture 1.

It is easy to check that f is monotone.

$$
\begin{aligned}
f(Y) = \{&\texttt{november}, \texttt{scotland}\} \cup \\
&\{\texttt{noSun} \mid (\texttt{arctic} \in Y) \land (\texttt{november} \in Y)\} \cup \\
&\{\texttt{sun} \mid (\texttt{australia} \in Y) \land (\texttt{november} \in Y)\} \cup \\
&\{\texttt{arctic} \mid \texttt{scotland} \in Y\} \cup Y
\end{aligned}
$$

The idea is that $f(Y)$ contains all the atomic facts in the program (in this case `november` and `scotland`) together with all atoms that can be derived from atoms in Y using a single rule of inference in the inference system of Lecture 1.

It is easy to check that f is monotone. (Exercise!)
(Remember that there is no negation in Prolog.)

$$
\begin{aligned}
f(Y) = {} & \{\texttt{november}, \texttt{scotland}\} \cup \\
& \{\texttt{noSun} \mid (\texttt{arctic} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
& \{\texttt{sun} \mid (\texttt{australia} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
& \{\texttt{arctic} \mid \texttt{scotland} \in Y\} \cup Y
\end{aligned}
$$

We calculate

$$
\begin{aligned}
f(\{\,\}) &= \\
f(f(\{\,\})) &= \\
f(f(f(\{\,\}))) &= \\
f(f(f(f(\{\,\})))) &=
\end{aligned}
$$

$$
\begin{aligned}
f(Y) = {} & \{\mathtt{november}, \mathtt{scotland}\} \cup \\
& \{\mathtt{noSun} \mid (\mathtt{arctic} \in Y) \wedge (\mathtt{november} \in Y)\} \cup \\
& \{\mathtt{sun} \mid (\mathtt{australia} \in Y) \wedge (\mathtt{november} \in Y)\} \cup \\
& \{\mathtt{arctic} \mid \mathtt{scotland} \in Y\} \cup Y
\end{aligned}
$$

We calculate

$$
\begin{aligned}
f(\{\,\}) &= \{\mathtt{november}, \mathtt{scotland}\} \\
f(f(\{\,\})) &= \\
f(f(f(\{\,\}))) &= \\
f(f(f(f(\{\,\})))) &=
\end{aligned}
$$

$$
\begin{aligned}
f(Y) \;=\; & \{\texttt{november}, \texttt{scotland}\} \cup \\
& \{\texttt{noSun} \mid (\texttt{arctic} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
& \{\texttt{sun} \mid (\texttt{australia} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
& \{\texttt{arctic} \mid \texttt{scotland} \in Y\} \cup Y
\end{aligned}
$$

We calculate

$$
\begin{aligned}
f(\{\,\}) &= \{\texttt{november}, \texttt{scotland}\} \\
f(f(\{\,\})) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}\} \\
f(f(f(\{\,\}))) &= \\
f(f(f(f(\{\,\})))) &=
\end{aligned}
$$

$$
\begin{aligned}
f(Y) = \{&\texttt{november}, \texttt{scotland}\} \cup \\
&\{\texttt{noSun} \mid (\texttt{arctic} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
&\{\texttt{sun} \mid (\texttt{australia} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
&\{\texttt{arctic} \mid \texttt{scotland} \in Y\} \cup Y
\end{aligned}
$$

We calculate

$$
\begin{aligned}
f(\{\,\}) &= \{\texttt{november}, \texttt{scotland}\} \\
f(f(\{\,\})) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}\} \\
f(f(f(\{\,\}))) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}, \texttt{noSun}\} \\
f(f(f(f(\{\,\})))) &=
\end{aligned}
$$

$$
\begin{aligned}
f(Y) = {} & \{\texttt{november}, \texttt{scotland}\} \cup \\
& \{\texttt{noSun} \mid (\texttt{arctic} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
& \{\texttt{sun} \mid (\texttt{australia} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
& \{\texttt{arctic} \mid \texttt{scotland} \in Y\} \cup Y
\end{aligned}
$$

We calculate

$$
\begin{aligned}
f(\{\,\}) &= \{\texttt{november}, \texttt{scotland}\} \\
f(f(\{\,\})) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}\} \\
f(f(f(\{\,\}))) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}, \texttt{noSun}\} \\
f(f(f(f(\{\,\})))) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}, \texttt{noSun}\}
\end{aligned}
$$

$$
\begin{aligned}
f(Y) = \{ & \texttt{november}, \texttt{scotland}\} \cup \\
& \{\texttt{noSun} \mid (\texttt{arctic} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
& \{\texttt{sun} \mid (\texttt{australia} \in Y) \wedge (\texttt{november} \in Y)\} \cup \\
& \{\texttt{arctic} \mid \texttt{scotland} \in Y\} \cup Y
\end{aligned}
$$

We calculate

$$
\begin{aligned}
f(\{\,\}) &= \{\texttt{november}, \texttt{scotland}\} \\
f(f(\{\,\})) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}\} \\
f(f(f(\{\,\}))) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}, \texttt{noSun}\} \\
f(f(f(f(\{\,\})))) &= \{\texttt{november}, \texttt{scotland}, \texttt{arctic}, \texttt{noSun}\} \\
&= f(f(f(\{\,\})))
\end{aligned}
$$

$$
\begin{aligned}
f(Y) = \{&\mathtt{november}, \mathtt{scotland}\} \cup \\
&\{\mathtt{noSun} \mid (\mathtt{arctic} \in Y) \wedge (\mathtt{november} \in Y)\} \cup \\
&\{\mathtt{sun} \mid (\mathtt{australia} \in Y) \wedge (\mathtt{november} \in Y)\} \cup \\
&\{\mathtt{arctic} \mid \mathtt{scotland} \in Y\} \cup Y
\end{aligned}
$$

We calculate

$$
\begin{aligned}
f(\{\,\}) &= \{\mathtt{november}, \mathtt{scotland}\} \\
f(f(\{\,\})) &= \{\mathtt{november}, \mathtt{scotland}, \mathtt{arctic}\} \\
f(f(f(\{\,\}))) &= \{\mathtt{november}, \mathtt{scotland}, \mathtt{arctic}, \mathtt{noSun}\} \\
f(f(f(f(\{\,\})))) &= \{\mathtt{november}, \mathtt{scotland}, \mathtt{arctic}, \mathtt{noSun}\} \\
&= f(f(f(\{\,\})))
\end{aligned}
$$

The set $f(f(f(\{\,\})))$ is the least fixed point of f.

Observations

The least fixed point

$$\{\texttt{november}, \texttt{scotland}, \texttt{arctic}, \texttt{noSun}\}$$

contains exactly the logical consequences of the program.

For any set Y of atoms define an interpretation $\mathcal{I}_Y$ by

$$\mathcal{I}_Y(q) \;=\; \begin{cases} \textbf{true} & \text{if } q \in Y \\ \textbf{false} & \text{if } q \notin Y \end{cases}$$

Then $\mathcal{I}_{\{\texttt{november},\texttt{scotland},\texttt{arctic},\texttt{noSun}\}}$ is a model of our program.

Another fixed point of f is

$$\{\texttt{november}, \texttt{scotland}, \texttt{arctic}, \texttt{noSun}, \texttt{australia}, \texttt{sun}\}$$

and $\mathcal{I}_{\{\texttt{november},\texttt{scotland},\texttt{arctic},\texttt{noSun},\texttt{australia},\texttt{sun}\}}$ is another model.

The general propositional case

We now consider a general program in definite clause propositional logic, given as a set P of axioms, each of the one of the forms

$$\begin{array}{rcl} & & q \\ p_1 \wedge \ldots \wedge p_k & \rightarrow & q \end{array}$$

Let X be the finite set of all atoms appearing in the program.

Define $f\colon \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ by

$$\begin{aligned} f(Y) = Y \; \cup \; & \{q \mid q \in P \text{ is an atom}\} \cup \\ & \{q \mid (p_1 \wedge \cdots \wedge p_k \rightarrow q) \in P \text{ and } p_1 \in Y \text{ and } \ldots \text{ and } p_k \in Y\} \end{aligned}$$

It is easy to check that f is monotone.

The general propositional case

We now consider a general program in definite clause propositional logic, given as a set P of axioms, each of the one of the forms

$$\begin{array}{rcl} & & q \\ p_1 \wedge \ldots \wedge p_k & \rightarrow & q \end{array}$$

Let X be the finite set of all atoms appearing in the program.

Define $f\colon \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ by

$$\begin{aligned} f(Y) = Y \ &\cup \ \{q \mid q \in P \text{ is an atom}\} \cup \\ &\{q \mid \ (p_1 \wedge \cdots \wedge p_k \rightarrow q) \in P \text{ and } p_1 \in Y \text{ and } \ldots \text{ and } p_k \in Y\} \end{aligned}$$

It is easy to check that f is monotone. (Exercise!)

Fixed points and models

$$\mathcal{I}_Y(q) \;=\; \begin{cases} \textbf{true} & \text{if } q \in Y \\ \textbf{false} & \text{if } q \notin Y \end{cases}$$

Theorem

$\mathcal{I}_Y$ is a model of the program P if and only if Y is a fixed point of the function f.

Theorem

If Y is the least fixed point of f then $\mathcal{I}_Y$ is the *minimal model*. That is, for any other model $\mathcal{I}'$, it holds that

$$\mathcal{I}_Y(q) = \textbf{true} \quad \text{implies} \quad \mathcal{I}'(q) = \textbf{true}$$

for all propositional atoms q.

Fixed points and models

$$\mathcal{I}_Y(q) \;=\; \begin{cases} \textbf{true} & \text{if } q \in Y \\ \textbf{false} & \text{if } q \notin Y \end{cases}$$

Theorem (Proof straightforward)
$\mathcal{I}_Y$ is a model of the program P if and only if Y is a fixed point of the function f.

Theorem (Proof straightforward)
If Y is the least fixed point of f then $\mathcal{I}_Y$ is the *minimal model*.
That is, for any other model $\mathcal{I}'$, it holds that

$$\mathcal{I}_Y(q) = \textbf{true} \quad \text{implies} \quad \mathcal{I}'(q) = \textbf{true}$$

for all propositional atoms q.

Fixed points and models

$$\mathcal{I}_Y(q) \;=\; \begin{cases} \textbf{true} & \text{if } q \in Y \\ \textbf{false} & \text{if } q \notin Y \end{cases}$$

Theorem (Proof straightforward)

$\mathcal{I}_Y$ is a model of the program P if and only if Y is a fixed point of the function f.

Theorem (Proof straightforward)

If Y is the least fixed point of f then $\mathcal{I}_Y$ is the *minimal model*. That is, for any other model $\mathcal{I}'$, it holds that

$$\mathcal{I}_Y(q) = \textbf{true} \quad \text{implies} \quad \mathcal{I}'(q) = \textbf{true}$$

for all propositional atoms q.

We now show that f always has a least fixed point and this gives us a means of constructing the minimal model.

Existence of least fixed point — finite case

Theorem

Suppose $f\colon \mathcal{P}(X) \to \mathcal{P}(X)$ is monotone, where X is a finite set. Define:

$$\begin{aligned} f^0(\{\,\}) &= \{\,\} \\ f^{n+1}(\{\,\}) &= f(f^n(\{\,\})) \end{aligned}$$

Then for some $N \leq |X|$ (where $|X|$ is the number of elements of X) it holds that $f^N(\{\,\})$ is the least fixed point of f.

In particular, every monotone function $f\colon \mathcal{P}(X) \to \mathcal{P}(X)$ has a least fixed point.

Note that one can think of $f^n(\{\,\})$ as

$$\overbrace{f(\,f(\ldots\,f(}^{n \text{ times}}\{\,\})\ldots))$$

Proof

We first prove, by induction on n that, $f^n(\{\}) \subseteq f^{n+1}(\{\})$.
When $n = 0$ we have $f^0(\{\}) = \{\} \subseteq f^1(\{\})$ because the empty set is a subset of every set.
When $n > 0$, the induction hypothesis gives that $f^{n-1}(\{\}) \subseteq f^n(\{\})$. So, by monotonicity, we have that $f(f^{n-1}(\{\})) \subseteq f(f^n(\{\}))$. That is, indeed $f^n(\{\}) \subseteq f^{n+1}(\{\})$.

So, for every $n \geq 0$, either $f^n(\{\})$ is a fixed point, or $f^{n+1}(\{\})$ contains at least one new element not contained in $f^n(\{\})$. Since there are only $|X|$ possible new elements to include, a fixed point must be reached at $f^N(\{\})$ for some $N \leq |X|$.

To show that $f^N(\{\})$ is the least fixed point, let Y be any fixed point. We show, by induction on n, that $f^n(\{\}) \subseteq Y$ always holds. This is trivial for $n = 0$. For $n > 0$, the induction hypothesis is that $f^{n-1}(\{\}) \subseteq Y$. By monotonicity, and because Y is a fixed point, $f(f^{n-1}(\{\})) \subseteq f(Y) = Y$. That is, $f^n(\{\}) \subseteq Y$ as required.

A decision procedure for propositional Prolog

Given a program P and a goal $q_1, \ldots, q_m$, we want to decide if $q_1 \wedge \cdots \wedge q_m$ is a logical consequence of P.

First construct the function f associated with P.

Successively calculate $f(\{\})$, $f(f(\{\}))$, $f(f(f(\{\})))$, ...

By the theorem, we know that we will encounter the least fixed point after at most $|X|$ steps, where X is the set of atoms in the program.

We detect when we have arrived at the least fixed point by checking if the next application of f leaves the set of atoms unchanged.

Now simply check if the all the goal atoms $q_1, \ldots, q_m$ are contained in the least fixed point. If so, return **yes**. Otherwise, return **no**.

The general (predicate logic) case

Now we consider the case of definite clause predicate logic, where the program P is a set $F_1, \ldots, F_n$ of definite clauses

$$B$$

$$A_1 \wedge \cdots \wedge A_k \ \rightarrow B$$

understood as implicitly universally quantified.

Herbrand models

A structure $\mathcal{S}$ is called a *Herbrand structure* if:

- The universe is the Herbrand universe.
- A constant $\mathtt{c}$ is interpreted by $\mathtt{c}^{\mathcal{S}} = \mathtt{c}$.
- A function symbol $\mathtt{f}/k$ is interpreted by $\mathtt{f}^{\mathcal{S}}(u_1, \ldots, u_k) = \mathtt{f}(u_1, \ldots, u_k)$.

A *Herbrand model* is just a Herbrand structure $\mathcal{S}$ that is a model of the program P.

A Herbrand model $\mathcal{S}$ is called *minimum* if, for any other Hebrand model $\mathcal{S}'$, it holds that

$$p^{\mathcal{S}}(u_1, \ldots, u_k) = \mathbf{true} \quad \text{implies} \quad p^{\mathcal{S}'}(u_1, \ldots, u_k) = \mathbf{true}$$

for every predicate symbol $\mathtt{p}/k$ and ground terms $u_1, \ldots, u_k$

Interpreting program as monotone function

A *ground atomic formula* is an atomic formula containing no variables.

We use the program P to define a function $f : \mathcal{P}(X) \to \mathcal{P}(X)$, where X is the set of all ground atomic formulas, by

$$
\begin{aligned}
f(Y) = Y \cup \{B\theta \mid B \in P \text{ is atomic }, \theta \text{ a ground substitution}\} \cup \\
\{B\theta \mid (A_1 \wedge \cdots \wedge A_k \to B) \in P, \theta \text{ a ground substitution}, \\
A_1\theta \in Y \text{ and } \ldots \text{ and } A_k\theta \in Y\}
\end{aligned}
$$

where a *ground substitution* is a substitution of ground terms to variables.

It is straightforward to show that f is monotone.

The definition is very similar to the propositional case. The main differences are: the use of substitutions, and the fact that the set X is not in general finite in the case of predicate logic.

Fixed points and Herbrand models

For a set Y of ground atomic formulas, define a Herbrand structure $\mathcal{H}_Y$ by:

$$p^{\mathcal{H}_Y}(u_1, \ldots, u_k) = \textbf{true} \quad \Leftrightarrow \quad p(u_1, \ldots, u_k) \in Y$$

Theorem

$\mathcal{H}_Y$ is a Herbrand model of the program P if and only if Y is a fixed point of the function f.

Theorem

If Y is the least fixed point of f then $\mathcal{H}_Y$ is the minimum Herbrand model.

As in the propositional case, we show that f always has a least fixed point and this gives us an alternative description of the Minimum Herbrand model (i.e., alternative to the description of the Minimum Herbrand model given in Lecture 5).

Existence of least fixed point — general case

The Knaster-Tarski Theorem

Suppose $f : \mathcal{P}(X) \to \mathcal{P}(X)$ is monotone. Then f has a least fixed point.

Proof

Define

$$Y = \bigcap \{Z \subseteq X \mid f(Z) \subseteq Z\}$$

We have that $f(Y) \subseteq Y$ because

$$f(Y) = f(\bigcap \{Z \mid f(Z) \subseteq Z\}) \subseteq \bigcap \{f(Z) \mid f(Z) \subseteq Z\}) \subseteq \bigcap \{Z \mid f(Z) \subseteq Z\}$$

using monotonicity for the middle inclusion.
Now, if W is any set satisfying $f(W) \subseteq W$, then $W \in \{Z \mid f(Z) \subseteq Z\}$, so

$$Y = \bigcap \{Z \mid f(Z) \subseteq Z\} \subseteq W$$

One such W is $W = f(Y)$ since, by monotonicity, $f(f(Y)) \subseteq f(Y)$. So $Y \subseteq f(Y)$.
Thus Y is a fixed point.

Also, any fixed point W satisfies $f(W) \subseteq W$, so $Y \subseteq W$. Thus indeed, Y is least.

Existence of least fixed point — general case

The Knaster-Tarski Theorem
Suppose $f\colon \mathcal{P}(X) \to \mathcal{P}(X)$ is monotone. Then f has a least fixed point.

Proof (non-examinable)
Define

$$Y = \bigcap \{Z \subseteq X \mid f(Z) \subseteq Z\}$$

We have that $f(Y) \subseteq Y$ because

$$f(Y) = f(\bigcap \{Z \mid f(Z) \subseteq Z\}) \subseteq \bigcap \{f(Z) \mid f(Z) \subseteq Z\}) \subseteq \bigcap \{Z \mid f(Z) \subseteq Z\}$$

using monotonicity for the middle inclusion.
Now, if W is any set satisfying $f(W) \subseteq W$, then $W \in \{Z \mid f(Z) \subseteq Z\}$, so

$$Y = \bigcap \{Z \mid f(Z) \subseteq Z\} \subseteq W$$

One such W is $W = f(Y)$ since, by monotonicity, $f(f(Y)) \subseteq f(Y)$. So $Y \subseteq f(Y)$. Thus Y is a fixed point.

Also, any fixed point W satisfies $f(W) \subseteq W$, so $Y \subseteq W$. Thus indeed, Y is least.

Main points today

Notions of monotone function, fixed point, least fixed point

Notion of Herbrand model and minimum Herbrand model

Interpreting program as a monotone function

Correspondence between (least) fixed points and (minimum) models

General existence of least fixed points, and proof in finite case

Decision procedure for propositional case by calculating fixed point