Computational Logic

Logic Programming:

Model and Fixpoint Semantics

Towards the Model and Fixpoint Semantics

- We have seen previously the operational semantics (SLD-resolution).
- We now present the (declarative) *Model Semantics*:
  - We define our semantic *domain* (Herbrand interpretations).
  - We introduce the Minimal Herbrand Model.
- And the (also declarative) *Fixpoint Semantics*.
  - We recall some basic fixpoint theory.
  - Present the T_P operator and the classic fixpoint semantics.

Declarative Semantics – Herbrand Base and Universe

- Given a first-order language L, with a non-empty set of variables, constants, function symbols, relation symbols, connectives, quantifiers, etc. and given a syntactic object A,

  $$ground(A) = \{A\theta | \exists\theta \in Subst, var(A\theta) = \emptyset\}$$

  i.e. the set of all "ground instances" of A.
- Given L, U_L *(Herbrand universe)* is the set of all ground terms of L.
- B_L *(Herbrand Base)* is the set of all ground atoms of L.
- Similarly, for the language L_P associated with a given program P we define U_P, and B_P.

Declarative Semantics – Herbrand Base and Universe (example)

- Program:

  $$\begin{aligned} P = \{ \ & p(f(X)) \leftarrow p(X). \\ & p(a). \\ & q(a). \\ & q(b). \ \} \end{aligned}$$

- Herbrand universe:
  $U_P = \{a, b, f(a), f(b), f(f(a)), f(f(b)), \ldots\}$
- Herbrand base:
  $B_P = \{p(a), p(b), q(a), q(b), p(f(a)), p(f(b)), q(f(a)), \ldots\}$

Herbrand Interpretations and Models

- A *Herbrand Interpretation* is a subset of B_L, i.e. the set of all Herbrand interpretations $I_L = \wp(B_L)$.
  (Note that I_L forms a *complete lattice* under $\subseteq$ – important for fixpoint operations to be introduced later).
- In previous example: $P = \{ \quad p(f(X)) \leftarrow p(X). \quad p(a). \quad q(a). \quad q(b). \quad \}$
  $U_P = \{a, b, f(a), f(b), f(f(a)), f(f(b)), \ldots\}$
  $B_P = \{p(a), p(b), q(a), q(b), p(f(a)), p(f(b)), q(f(a)), \ldots\}$
  $I_P =$ *all subsets of* B_P
- A *Herbrand Model* is a Herbrand interpretation which contains all logical consequences of the program.
- The *Minimal Herbrand Model* H_P is the smallest Herbrand interpretation which contains all logical consequences of the program. (Theorem: it is unique.)
- Example:
  $H_P = \{q(a), q(b), p(a), p(f(a)), p(f(f(a))), \ldots\}$

Declarative Semantics, Completeness, Correctness

- *Declarative semantics of a logic program* P:
  the set of ground facts which are logical consequences of the program (i.e., H_P).
  (I.e., the *Minimal Herbrand* model (or "least model") of P).
- *Intended meaning of a logic program* P:
  the set I of ground facts that the user expects to be logical consequences of the program.
- A logic program is *correct* if $H_P \subseteq I$.
- A logic program is *complete* if $I \subseteq H_P$.
- Example:

  father(john,peter).

  father(john,mary).

  mother(mary,mike).

  grandfather(X,Y) $\leftarrow$ father(X,Z), father(Z,Y).

  with the usual intended meaning is *correct* but *incomplete*.

Towards a Fixpoint Semantics for LP – Fixpoint Basics

- A *fixpoint* for an operator $T : X \rightarrow X$ is an element of $x \in X$ such that $x = T(x)$.
- If X is a poset, T is monotonic if $\forall x, y \in X,\ x \leq y \Rightarrow T(x) \leq T(y)$
- If X is a complete lattice and T is monotonic the set of fixpoints of T is also a complete lattice [Tarski]
- The least element of the lattice is the *least fixpoint* of T, denoted $lfp\ (T)$
- Powers of a monotonic operator (successive applications):

  $T \uparrow 0(x) = x$
  $T \uparrow n(x) = T(T \uparrow (n-1)(x))(n$ is a successor ordinal$)$
  $T \uparrow \omega(x) = \sqcup\{T \uparrow n(x) | n < \omega\}$

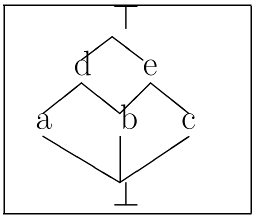
  We abbreviate $T \uparrow \alpha(\perp)$ as $T \uparrow \alpha$
- There is some ω such that $T \uparrow \omega = lfp\ T$. The sequence $T \uparrow 0, T \uparrow 1, ..., lfp\ T$ is the *Kleene sequence* for T
- In a finite lattice the Kleene sequence for a monotonic operator T is finite

Towards a Fixpoint Semantics for LP – Fixpoint Basics (Contd.)

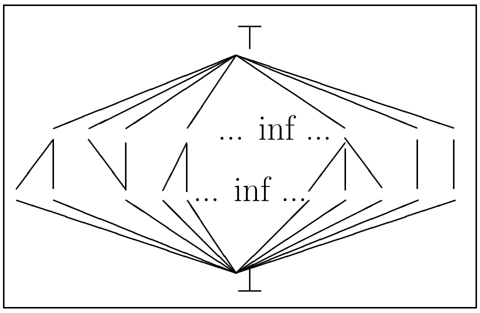
- A subset Y of a poset X is an (ascending) chain iff $\forall y, y' \in Y, y \leq y' \vee y' \leq y$
- A complete lattice X is *ascending chain finite* (or *Noetherian*) if all ascending chains are finite
- In an ascending chain finite lattice the Kleene sequence for a monotonic operator T is finite

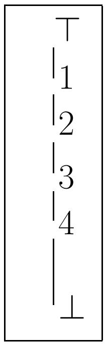
Lattice Structures

finite



finite_depth





ascending chain finite

A Fixpoint Semantics for Logic Programs

- Semantic *domain*: $I_L = \wp(B_L)$.
- I.e., the elements of the semantic domain and *interpretations* (subsets of the Herbrand base).
- Semantic *operator* (defined on programs): the *immediate consequences operator*, T_P:
  - ◇ T_P is a mapping: $T_P : I_P \rightarrow I_P$ defined by:

    $T_P(I) = \{A \in B_P \mid \exists C \in ground(P), C = A \leftarrow L_1, ..., L_n$ and $L_1, \ldots L_n \in I\}$

  (in particular, if $(A \leftarrow) \in P$, then every element of $ground(A)$ is in $T_P(I)$, $\forall\ I$).
- T_P is monotonic, so:
  - ◇ it has a least fixpoint I^* so that $T_P(I^*) = I^*$,
  - ◇ this fixpoint can be obtained by applying T_P iteratively starting from the bottom element of the lattice (the empty interpretation).

A Fixpoint Semantics for Logic Programs: Example 1 (finite)

$P = \{\ p(X, a) \leftarrow q(X).$
$\quad p(X, Y) \leftarrow q(X), r(Y).$
$\quad q(a). \quad r(b).$
$\quad q(b). \quad r(c).\ \}$

$U_P = \{a, b, c\}$

$B_P = \{\ p(a, a), p(a, b), p(a, c), p(b, a), p(b, b), p(b, c), p(c, a), p(c, b), p(c, c),$
$\quad q(a), q(b), q(c),$
$\quad r(a), r(b), r(c)\}$

$I_P =$ all subsets of B_P
$H_P = \{q(a), q(b), r(b), r(c), p(a, a), p(b, a), p(a, b), p(b, b), p(a, c), p(b, c)\}$

$T_P \uparrow 0 = \{q(a), q(b), r(b), r(c)\}$
$T_P \uparrow 1 = \{q(a), q(b), r(b), r(c)\} \cup \{p(a, a), p(b, a), p(a, b), p(b, b), p(a, c), p(b, c)\}$
$T_P \uparrow 2 = T_P \uparrow 1 = lfp(T_P) = H_P$

A Fixpoint Semantics for Logic Programs: Example 2 (infinite)

$P = \{\ p(f(X)) \leftarrow p(X).$
$\quad p(a).$
$\quad q(a).$
$\quad q(b).\ \}$

$U_P = \{a, b, f(a), f(b), f(f(a)), f(f(b)), \ldots\}$
$B_P = \{p(a), p(b), q(a), q(b), p(f(a)), p(f(b)), q(f(a)), \ldots\}$
$I_P =$ *all subsets of* B_P
$H_P = \{q(a), q(b), p(a)\} \cup \{p(f^n(a)) \mid n \in \mathcal{N}\}$
where we define $f^n(a)$ to be f nested n times and then applied to a.
(i.e., $q(a), q(b), p(a), p(f(a)), p(f(f(a))), p(f(f(f(a)))),$ ...)

$T_P \uparrow 0 = \{p(a), q(a), q(b)\}$
$T_P \uparrow 1 = \{p(a), q(a), q(b), p(f(a))\}$
$T_P \uparrow 2 = \{p(a), q(a), q(b), p(f(a)), p(f(f(a)))\}$
$\ldots$
$T_P \uparrow \omega = H_P$

A Fixpoint Semantics for Logic Programs: Example 3 (infi nite)

- Example:

$$
\begin{aligned}
P = \{ \; & nat(0). \\
& nat(s(X)) \leftarrow nat(X). \\
& \\
& sum(0, X, X). \\
& sum(s(X), Y, s(Z)) \leftarrow sum(X, Y, Z). \; \}
\end{aligned}
$$

$U_P = \{0\} \cup \{s(x) \mid x \in U_P\}$

(i.e., $\{0, s(0), s(s(0)), s(s(s(0))), ...\}$).

$B_P = \{nat(x) \mid x \in U_P\} \cup \{sum(x, y, z) \mid x, y, z \in U_P\}$

(i.e., $\{nat(0), nat(s(0)), nat(s(s(0))), ...\} \cup$
$\{sum(0, 0, 0), sum(s(0), 0, 0), sum(0, s(0), 0), sum(0, 0, s(0)), ...\}$).

A Fixpoint Semantics for Logic Programs: Example 3 (infi nite, cont.)

Constructing the least fixpoint of the T_P operator:

$$
\begin{aligned}
T_P \uparrow 0 &= \{nat(0)\} \cup \{sum(0, x, x) \mid x \in U_P\} \\
T_P \uparrow 1 &= T_P \uparrow 0 \cup \{nat(s(0))\} \\
&\quad \cup \{sum(s(0), y, s(y)) \mid y \in U_P\} \\
T_P \uparrow 2 &= T_P \uparrow 1 \cup \{nat(s(s(0)))\} \\
&\quad \cup \{sum(s(s(0)), y, s(s(y))) \mid y \in U_P\} \\
T_P \uparrow 3 &= T_P \uparrow 2 \cup \{nat(s(s(s(0))))\} \\
&\quad \cup \{sum(s(s(s(0))), y, s(s(s(y)))) \mid y \in U_P\} \\
&\ldots \\
T_P \uparrow \omega &= \{nat(x) \mid x \in U_P\} \cup \\
&\quad \{sum(s^n(0), y, s^n(y)) \mid y \in U_P \wedge n \in \mathcal{N}\}
\end{aligned}
$$

where we define $s^x(y)$ to be s nested x times and then applied to y.

Semantics – Equivalences

- (Characterization Theorem) [Van Emden and Kowalski]
  A program P has a Herbrand model H_P such that :
  - H_P is the least Herbrand Model of P.
  - H_P is the least fixpoint of T_P ($\mathit{lfp}\ T_P$).
  - $H_P = T_P \uparrow \omega$.

  I.e., *least model semantics* (H_P) $\equiv$ *fixpoint semantics* ($\mathit{lfp}\ T_P$)
- In addition, there is also an equivalence with the *operational semantics* (SLD-resolution):
  - SLD-resolution answers “yes” to $a \in B_P \iff a \in H_P$.
- Because it gives us a way to directly build H_P (for finite models), the least fixpoint semantics can in some cases also be an operational semantics (e.g., for *datalog* in *deductive databases*).