Characterizing Classes of Logic Programs via Unique Fixed-Points of Monotonic Operators

Pascal Hitzler and Anthony Karel Seda

National University of Ireland, Cork

Contents

Slides for presentation at the Workshop on Domains 5 in Darmstadt, Germany.

Logic Programs and Models

A (normal) logic program P is

a finite set of clauses of the form ($m \geq 0$)

$$\forall(\underbrace{A}_{\texttt{head}} \leftarrow \underbrace{L_1 \wedge \cdots \wedge L_m}_{\texttt{body}})$$

* A atom
* L_i literals

Written: $A \leftarrow L_1, \ldots, L_m$

* B_P set of all ground atoms in P
* $I_P = 2^{B_P}$ set of all interpretations for P
* ground(P) set of ground instances of clauses in P

models give declarative semantics
a program may have many models: *intended* model?
many approaches to choosing a semantics exist
we focus on *supported models*

single-step operator on I_P (in general not monotonic):

$$T_P(I) = \{A \in B_P \mid \text{ there exists } \texttt{A} \leftarrow \texttt{body} \in \text{ground}(P) \text{ with } I \models \texttt{body}\}$$

* I model iff $T_P(I) \subseteq I$ (pre-fixed point)
* I *supported* model iff $T_P(I) = I$ (fixed point)

Generalized Ultrametric Spaces; The Prieß-Crampe & Ribenboim Theorem

X set, Γ poset, $\min \Gamma = 0$.
$d : X \times X \to \Gamma$ is *gum* space iff $\forall x, y, z \in X, \gamma \in \Gamma$

- $d(x, y) = 0$ iff $x = y$
- $d(x, y) = d(y, x)$
- $d(x, y) \leq \gamma$ and $d(y, z) \leq \gamma \Longrightarrow d(x, z) \leq \gamma$

(X, d) *spherically complete* iff $\bigcap \mathcal{C} \neq \emptyset$ for each chain $\mathcal{C}$ of *balls* ($B_\gamma(y) = \{x \mid d(x, y) \leq \gamma\}$) in X.

Theorem (P-C & R)

(X, d) spherically complete gum space
$f : X \to X$ *contraction*

$$(d(f(x), f(y)) < d(x, y) \qquad \forall x, y \in X).$$

Then f has a unique fixed point.

Domains as Gums

D algebraically complete cpo (e.g. I_P)
γ countable ordinal, $\Gamma_\gamma = \{2^{-\alpha} \mid \alpha < \gamma\}$
$2^{-\alpha} < 2^{-\beta}$ iff $\beta < \alpha$
$r : D_C \to \gamma + 1$ *rank function*
$d_r : D \times D \to \Gamma_{\gamma+1}$ defined by

$$d_r(x, y) = \inf\{2^{-\alpha} \mid (c \sqsubseteq x \text{ iff } c \sqsubseteq y) \text{ for all } c \in D_C \text{ with } r(c) < \alpha\}$$

Theorem Hitzler & Seda 1999

(D, d_r) is a spherically complete gum space.

Proof uses the following Lemma.
Let $B_{2^{-\beta}}(y)$ (=: $B_\beta(y)$) be a ball in (D, d_r).

- ∘ $x \in B_\beta(y) \Longrightarrow \{c \in \text{approx}(x) \mid r(c) < \beta\}$
$$= \{c \in \text{approx}(y) \mid r(c) < \beta\}$$
- ∘ $B_\beta = \sup\{c \in \text{approx}(y) \mid r(c) < \beta\}$ exists
- ∘ $B_\beta \in B_\beta(y)$
- ∘ $B_\alpha(x) \subseteq B_\beta(y) \Longrightarrow B_\beta \sqsubseteq B_\alpha$

* generalizes earlier result Seda & Hitzler 1997
* P-C & R Theorem is more general than applied
* bottom element of D not needed
* can replace $\Gamma_{\gamma+1}$ with totally ordered set with infs

Application: lh-programs

Logic program P *locally hierarchical* (Cavedon)
(S & H: *strictly level-decreasing*) iff
exists *level mapping* $l : B_P \to \gamma$ ($l(A) = l(\neg A)$) s.t.
for all $A \leftarrow \texttt{body} \in \text{ground}(P)$
$l(A) > l(L)$ for all L in $\texttt{body}$.

Set $r(I) = \max\{l(A) \mid A \in I\}$ for finite $I \in I_P$.

Theorem

If P lh then T_P contraction (wrt. (D, d_r)). Hence, P has unique supported model.

3-valued Interpretations and Operators

Truth Values: t, f, u
$I = (I^+, I^-)$, $I^+, I^- \in I_P$ with $I^+ \cap I^- = \emptyset$
3-valued Interpretaion
$A \in I^+$ are true, $B \in I^-$ are false, others are undefined.
I called *total* if $I^+ \cup I^- = B_P$.
$I_{P,3}$ set of all 3-valued interpretations
$I_{P,3}$ is cpo (in fact, complete semilattice, Fitting 1985):

$$I \leq J \text{ iff } I^+ \subseteq J^+ \text{ and } I^- \subseteq J^-$$

Program Transformation:
A occurring as head in ground(P),
form *pseudo clause* $A \leftarrow \bigvee_i C_i$
body is disj. of bodies C_i of all clauses with head A.
resulting set of pseudo clauses is denoted P^*.

Monotonic Operator F on $I_{P,3}$
$*$ has least fixed point $F\uparrow\alpha$, α ordinal.
$F\uparrow 0 = (\emptyset, \emptyset)$
$F\uparrow(\alpha+1) = F(F\uparrow\alpha)$
$F\uparrow(\alpha) = \bigcup_{\beta<\alpha} F\uparrow\beta$ for α limit ordinal

Least fixed point (lfp) is maximal in $I_{P,3}$
iff it is a total 3-valued interpretation.

Choice of evaluating logical connectives:
Negation: $\neg t = f$, $\neg f = t$, $\neg u = u$
Conjunction and Disjunction (extend to pseudo-clauses):

		C_1	C_2	C_3	D_1	D_2
p	q	$p \wedge q$	$p \wedge q$	$p \wedge q$	$p \vee q$	$p \vee q$
t	t	t	t	t	t	t
t	u	u	u	u	t	u
t	f	f	f	f	t	t
u	t	u	u	u	t	u
u	u	u	u	u	u	u
u	f	f	u	u	u	u
f	t	f	f	f	t	t
f	u	f	f	u	u	u
f	f	f	f	f	f	f

C_1, D_1: Fitting's Kripke-Kleene Semantics 1985, Mycroft 1984, Kunen 1987,Apt & Pedreschi 1993, Naish 1998.
C_2, D_1: Barbuti et al. 1991, Andrews 1997.

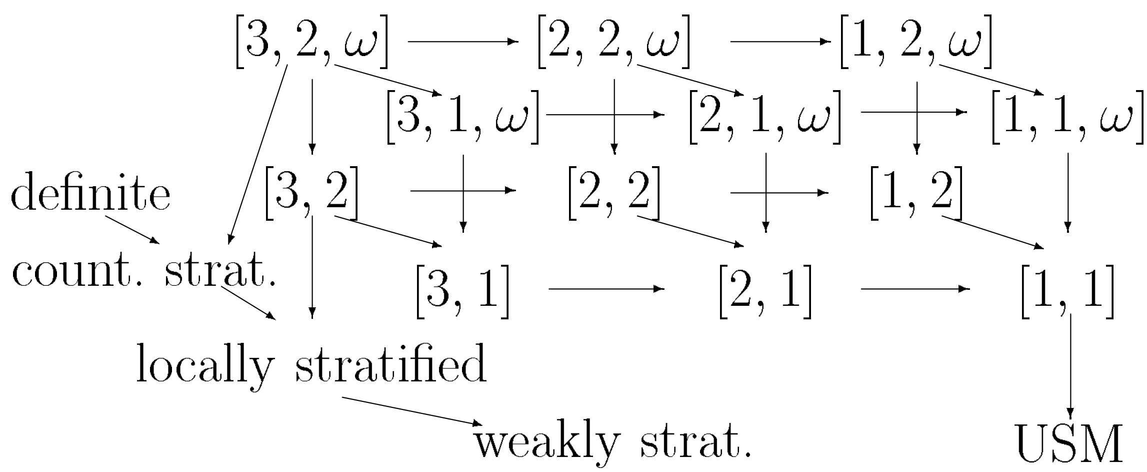
Operators on $I_{P,3}$ using P^*
$\Phi_{i,j}$, $i = 1, 2, 3$ conjunction, $j = 1, 2$ disjunction
$\Phi_{i,j}(I) = (T, F)$ with
$T = \{A \in B_P \mid \exists(\texttt{head} \leftarrow \texttt{body}),\ \texttt{body} \text{ is true in } I\}$
$F = \{A \in B_P \mid \forall(\texttt{head} \leftarrow \texttt{body}),\ \texttt{body} \text{ is false in } I\}$

Unique Supported Model Classes

$\Phi_{i,j}$ monotonic for all choices of i, j.
Define **classes** of programs:
$[i, j]$ all programs s.t. lfp of $\Phi_{i,j}$ is total
$[i, j, \omega]$ all progs. s.t. lfp of $\Phi_{i,j}$ is $\Phi_{i,j}\!\uparrow\!\omega$ and is total

$\Phi_{i,j}$ has total lfp $I \Longrightarrow I^+$ is unique fixed point of T_P
$\Longrightarrow I^+$ is unique supported model.
I.e. $[i, j]$, $[i, j, \omega]$ are *unique supported model classes.*

Dependencies between the classes



$\longrightarrow$: set inclusion

Characterizations of Programs

$[3, 2, \omega]$	$[3, 2]$	$[2, 2, \omega]$	$[1, 2]$
acyclic	lh	acceptable	Φ^*-accessible

SLDNF-resolution and -trees

For simplicity: consider ground programs and goals.

P program, $Q = (\leftarrow L_1, \ldots, L_n)$ goal
Construct SLDNF-tree:
Choose selection function: selects one L_i for evaluation.

$L_i = A$ — **atom**
Daughters of Q are constructed as follows:
For each $A \leftarrow \texttt{body}$ which is clause in P,
$L_1, \ldots, L_{i-1}, \texttt{body}, L_{i+1}, \ldots, L_n$ is daughter of Q.
Leaves are empty goals (successful branch)
or when selected literal does not produce daughters
(branch fails).

$L_i = \neg A$ — **negative literal**
Construct SLDNF-tree from goal $\leftarrow A$.
If all branches of tree successful then Q fails.
Otherwise daugher of Q is $L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_n$.

Two decisions:
1. selection function
2. how to traverse tree

Prolog:
selects leftmost (new) literal + left-depth-first search

Problem: can not work on non-ground selected literals
(floundering)

Results on Program Classes

Acyclic Programs

Cavedon 1989: ω-*locally hierarchical programs*
Definition as for lh programs, but
level mapping maps into ω.
subclass of lh programs
Bezem 1989: exactly the terminating programs
i.e. all SLDNF-trees of ground goals are finite.
Bezem 1989: compute all total computable functions.

Locally Hierarchical Programs

Cavedon 1989
may not terminate
Seda & Hitzler 1998:
* compute all partial recursive functions
if use of cut-operator is allowed
(cut operator prunes SLDNF-tree)
* connections to topological dynamics
* topological constructions of unique supported model

Acceptable Programs

Definition Apt & Pedreschi 1993

Let p, q be predicate symbols in P.
p *refers to* q if there is a clause in P with p in its head and q in its body.
p *depends on* q if (p, q) is in the reflexive, transitive closure of the relation *refers to*.
Set of predicate symbols in P which occur in a negative literal in the body of a clause in P is denoted by Neg_P.
Set of predicate symbols in P on which the predicate symbols in Neg_P depend is denoted by Neg_P^*.
Define P^- to be the set of clauses in P which contain a predicate symbol from Neg_P^* in the head.

Program P is *acceptable* if there is
a level mapping l which maps into ω
and a supported model I of P^- s.t.
for every clause $A \leftarrow L_1, \ldots, L_n$ in ground(P)
whenever $I \models L_1 \wedge \cdots \wedge L_{i-1}$ we have $l(A) > l(L_i)$.

Apt & Pedreschi 1993:
* acceptable programs are left-terminating
(SLDNF-trees of ground goals are finite under leftmost selection rule)
* left-term. non-floundering programs are acceptable
* superclass of acyclic programs

they can not compute all computable functions
H & S 1999: topological characterization

Φ^*-Accessible Progams

Seda & Hitzler 1999:
* possible to characterize similar to acceptable prog.
* can compute all partial recursive functions
* superclass of lh and acceptable programs

Work in Progress

topological aspects:
* for usm classes transfinite iterates of T_P operator converge in Cantor topology
* topological characterizations of the classes

termination:
* non-commutative disjunction etc. should correspond to other strategies of traversing the SLDNF-tree

program classes:
* understanding the "space" of all programs
* understanding *all* usm programs

denotational semantics:
* adjust approach for other semantics