Towards a Semantical Hierarchy of Logic Programming Classes

Pascal Hitzler

(partially joint work with Anthony Karel Seda)

Dresden, April 2001

Knowledge Representation and Reasoning Group
Artificial Intelligence Institute
Department of Computer Science
Dresden University of Technology, Germany
phitzler@inf.tu-dresden.de
http://www.wv.inf.tu-dresden.de/~pascal/

Logic programs

A (*normal logic*) *program* P is a finite set of *clauses*

$$\forall(A \leftarrow L_1, \dots, L_n),$$

over a first-order language $\mathcal{L}$, written as

$$A \leftarrow L_1, \dots, L_n,$$

where A is an atom and the L_i are literals. (A *head*, $L_1, \dots, L_n$ *body*.)

P is called *definite*, if P does not contain negation symbols.

B_P: Herbrand base (set of all ground instances of atoms).
$I_P = 2^{B_P}$: set of all (2-valued) interpretations (complete lattice wrt. $\subseteq$).
ground(P): set of all ground instances of clauses from P.

Slide 3

Locally hierarchical/acyclic programs

P *locally hierarchical* (*lh*) (Cavedon 1989)
if exists *level mapping* $l : B_P \to \alpha$ for some ordinal α such that

$$l(A) > l(L_i)$$

for all $A \leftarrow L_1, \dots, L_n \in \text{ground}(P)$.

P *acyclic* if P lh and $\alpha = \omega \qquad (= \mathbb{N})$.

Immediate consequence operator $T_P : I_P \to I_P$: $T_P(I)$ is set of all $A \in B_P$ for which exists $A \leftarrow \texttt{body} \in \text{ground}(P)$ s.t. $I \models \texttt{body}$.

Fixed points of T_P are exactly the *supported models* of P.

Slide 4

Acceptable programs I (P^-)

P program, p, q predicate symbols in P.
p *refers to* q iff exists ground clause with p in head and q in body.
p *depends on* q is reflexive transitive closure of "refers to".
Neg_P set of all predicate symbols which occur negatively in P.
Neg_P^* set of all predicate symbols on which the predicate symbols in Neg_P depend.
P^- set of all clauses in ground(P) with head in Neg_P^*.

Acceptable programs II

P *acceptable* (Apt and Pedreschi 1993) iff
exists level mapping $l : B_P \to \omega$
and model I which is a supported model of P^-
(when restricted to atoms in Neg_P^*)
s.t. for all $A \leftarrow L_1, \ldots, L_n \in \mathrm{ground}(P)$
and all $i = 1, \ldots, i$ we have

$$I \models L_1 \wedge \cdots \wedge L_{i-1} \quad \text{implies} \quad l(A) > l(L_i).$$

Fitting operators

P program.
For each $A \in B_P$ form *pseudo clause*

$$A \leftarrow \bigvee C_i \quad (= \mathtt{body}_A)$$

where C_i are exactly the bodies of the clauses in ground(P) with head A.

Choose your favourite (suitable) many-valued logic Λ
with space of interpreatations I_Λ
and associate an operator $\Phi_{P,\Lambda} : I_\Lambda \to I_\Lambda$ by

$$\Phi_{P,\Lambda}(I)(A) = I\left(\mathtt{body}_A\right).$$

Slide 7

3-valued interpretations

3 truth values $\{f, u, t\}$ (false, undefined, true).
Set of all interpretations $I_{P,3}$ is set of pairs (T, F)
with $T, F \subseteq B_P$ and $T \cap F = \emptyset$.
T true atoms, F false atoms, rest undefined

$(T, F) \leq (T', F')$ iff $T \subseteq T'$ and $F \subseteq F'$.

$I_{P,3}$ is complete semilattice.

Fitting operators in the following logics are monotonic, i.e. have least fixed points. With each operator we associate a class of programs determined by the fact that the considered Fitting operator has a least fixed point which leaves nothing undefined.

Slide 8

Φ-accessible programs I

		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 associated class: Φ-*accessible programs*.
This is Kleene's strong 3-valued logic
and the original Fitting semantics (Fitting 1985).

Slide 9

Other classes

C_3, D_2 associated class: locally hierarchical programs
acyclic if least fixed point is reached at $\Phi\!\uparrow\!\omega$.

C_2, D_2 ssociated class: acceptable programs
if least fixed point if reached at $\Phi\!\uparrow\!\omega$.

C_1, D_2 associated class: Φ^*-*accessible* programs.
(This class is computationally universal)

P is Φ^*-accessible iff
for each $A \leftarrow L_1, \ldots, L_n$ in ground(P) either
$I \models L_1 \wedge \cdots \wedge L_n$ and $L(A) > l(L_i)$ for all i or
exists i s.t. $I \not\models L_i$, $I \not\models A$ and $l(A) > l(L_i)$.

Slide 10

Φ-accessible programs II

Alternative characterization

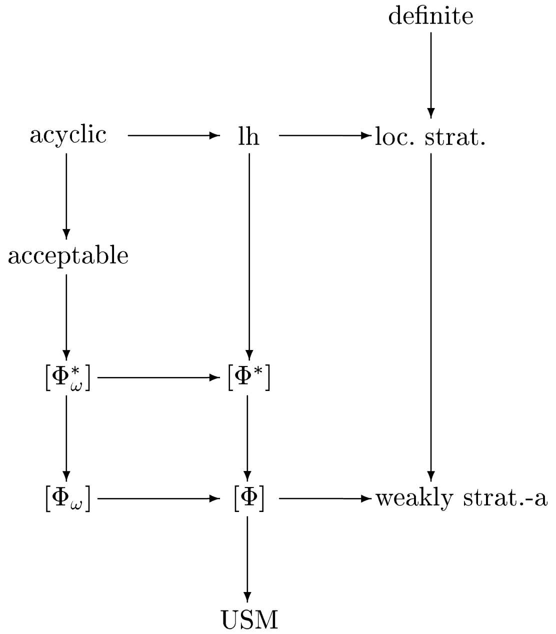
P is Φ-accessible iff
each $A \in B_P$ satisfies either (i) or (ii).

(i) Exists $A \leftarrow L_1, \ldots, L_n$ in ground(P) s.t.
$I \models L_1 \wedge \cdots \wedge L_n$ and $l(A) > l(L_i)$ for all i.

(ii) For each $A \leftarrow L_1, \ldots, L_n$ in ground(P)
exists i with $I \not\models L_i$, $I \not\models A$, $l(A) > l(L_i)$.

Slide 11

Diagram



stratified: Przymusinski 1988
weakly stratified:
Przymusinsy and Przymusinska 1990

Slide 12

Stable and Supported Models

$I \in I_P$ is *well-supported* if
exists strict well-founded partial order $\prec$ on I
s.t. for any $A \in I$ exists
$A \leftarrow B_1, \ldots, B_n, \neg C_1, \ldots, \neg C_m$ in ground(P)
with $I \models B_1 \wedge B_n \wedge \neg C_1 \wedge \cdots \wedge \neg C_m$
and $B_i \prec A$ for each i.
(Fages 1991, 1994)

Let P be a logic program.
M is well-supported model iff M is stable model.

Let P' be obtained from P by omitting
all negative literals in the clauses.

Let P be program s.t. P' is Φ^*-accessible.
Then M supported model iff M stable model.

$*$ Result does not generalize to $[\Phi]$.