Nonmonotonic logics and their algebraic foundations

Mirosław Truszczyński

Department of Computer Science
University of Kentucky

Computer Science Logic 2006
Szeged, September 25-29, 2006

Plan

- Beginnings of nonmon logics
- General overview of the field
- Brief comments on nonmon inference relations, preference logics, and preferential models
- Main focus: logics defining belief sets through fixpoint conditions
  - particularly, their abstract algebraic foundations
  - and what algebra buys you
- Concluding remarks

McCarthy and Hayes on AI, 1969

[...] intelligence

- has two parts, which we shall call the epistemological and the heuristic.
  The epistemological part is the representation of the world in such a form that the solution of problems follows from the facts expressed in the representation. The heuristic part is the mechanism that on the basis of the information solves the problem and decides what to do.

Epistemological part → knowledge representation

- Obvious approach (McCarthy): use FOL logic

McCarthy and Hayes on AI, 1969

[...] intelligence

- has two parts, which we shall call the epistemological and the heuristic.
  The epistemological part is the representation of the world in such a form that the solution of problems follows from the facts expressed in the representation. The heuristic part is the mechanism that on the basis of the information solves the problem and decides what to do.

Epistemological part → knowledge representation

- Obvious approach (McCarthy): use FOL logic

Knowledge representation through classical logic

It is not so simple

- Qualification problem
  (we do not check for potato in tailpipe before starting the engine)
- Frame problem
  (moving an object does not change its color)
- Rules with exceptions (defaults)
- Negative information

University-professor example

Basic scenario

- Professors teach
- Department chairs are professors
- Dr. Jones is a professor
- Thus, Dr. Jones teaches

Exception to a general rule

- Department chairs do not teach
- Dr. Jones is department chair
- Thus, Dr. Jones does not teach

University-professor example

Basic scenario

- Professors teach
- Department chairs are professors
- Dr. Jones is a professor
- Thus, Dr. Jones teaches

Exception to a general rule

- Department chairs do not teach
- Dr. Jones is department chair
- Thus, Dr. Jones does not teach

New information invalidates earlier inferences

Problem for classical logic!

- Classical logic is monotone: if $T \models \alpha$ and $T \subseteq T'$, then $T' \models \alpha$
- Direct representations do not work
  - $\mathit{prof}(X) \rightarrow \mathit{teaches}(X)$
  - $\mathit{chair}(X) \rightarrow \mathit{prof}(X)$
  - $\mathit{chair}(X) \rightarrow \neg\mathit{teaches}(X)$

More complex solutions necessary

- For instance:
  - $\mathit{prof}(X) \rightarrow \mathit{normally\_teaches}(X)$
  - $\mathit{chair}(X) \rightarrow \mathit{prof}(X)$
  - $\mathit{normally\_teaches}(X) \wedge \neg\mathit{chair}(X) \rightarrow \mathit{teaches}(X)$
  - $\mathit{chair}(X) \rightarrow \neg\mathit{teaches}(X)$

New information invalidates earlier inferences

Problem for classical logic!

- Classical logic is monotone: if $T \models \alpha$ and $T \subseteq T'$, then $T' \models \alpha$
- Direct representations do not work
  - $\mathit{prof}(X) \rightarrow \mathit{teaches}(X)$
  - $\mathit{chair}(X) \rightarrow \mathit{prof}(X)$
  - $\mathit{chair}(X) \rightarrow \neg \mathit{teaches}(X)$

More complex solutions necessary

- For instance:
  - $\mathit{prof}(X) \rightarrow \mathit{normally\_teaches}(X)$
  - $\mathit{chair}(X) \rightarrow \mathit{prof}(X)$
  - $\mathit{normally\_teaches}(X) \wedge \neg \mathit{chair}(X) \rightarrow \mathit{teaches}(X)$
  - $\mathit{chair}(X) \rightarrow \neg \mathit{teaches}(X)$

Nonmon logics — response to challenges of KR

What is it all about?

- Nonmonotonic inference
- Belief-set formation

Study of inference relations

Classical example — entailment relation

- Fix W — a set of propositional interpretations
- Define relation $\mathrel{|\!\sim}_W$:

$\alpha \mathrel{|\!\sim}_W \beta$ if β holds in every interpretation in W in which α holds

Study of inference relations

Specifying inference relation $\mid\!\sim$ through postulates

Monotony	if $\alpha \supset \beta$ is a tautology and $\beta \mid\!\sim \gamma$, then $\alpha \mid\!\sim \gamma$
Right Weakening	if $\alpha \supset \beta$ is a tautology and $\gamma \mid\!\sim \alpha$, then $\gamma \mid\!\sim \beta$
Reflexivity	$\alpha \mid\!\sim \alpha$
And	if $\alpha \mid\!\sim \beta$ and $\alpha \mid\!\sim \gamma$ then $\alpha \mid\!\sim \beta \wedge \gamma$
Or	if $\alpha \mid\!\sim \gamma$ and $\beta \mid\!\sim \gamma$ then $\alpha \vee \beta \mid\!\sim \gamma$

Characterize inference relations $\mid\!\sim_W$

Kraus, Lehmann, Magidor 1990

Study of inference relations

Specifying inference relation $\mid\sim$ through postulates

Monotony	if $\alpha \supset \beta$ is a tautology and $\beta \mid\sim \gamma$, then $\alpha \mid\sim \gamma$
Right Weakening	if $\alpha \supset \beta$ is a tautology and $\gamma \mid\sim \alpha$, then $\gamma \mid\sim \beta$
Reflexivity	$\alpha \mid\sim \alpha$
And	if $\alpha \mid\sim \beta$ and $\alpha \mid\sim \gamma$ then $\alpha \mid\sim \beta \wedge \gamma$
Or	if $\alpha \mid\sim \gamma$ and $\beta \mid\sim \gamma$ then $\alpha \vee \beta \mid\sim \gamma$

Characterize inference relations $\mid\sim_W$

Kraus, Lehmann, Magidor 1990

Study of nonmon inference relations

Preferential models

- A possible-world structure is a pair $\mathcal{W} = \langle W, v \rangle$
  - W — a set of *worlds*
  - v — a function mapping worlds to interpretations
  - $\mathcal{W}(\alpha) = \{w \in W : v(w) \models \alpha\}$
- A preferential model — a pair $\langle \mathcal{W}, \prec \rangle$
  - $\mathcal{W}$ — a possible-world structure
  - $\prec$ is a strict partial order on the worlds of W satisfying the smoothness condition
- $\alpha \mathrel{|\!\!\sim}_{\langle \mathcal{W}, \prec \rangle} \beta$ if β holds in every $\prec$-minimal world in $\mathcal{W}(\alpha)$.
- Preferential inference relations
- Do not obey *Monotony*
- Generalization of circumscription and preference logics

  McCarthy 1977, Shoham 1987, respectively

Study of nonmon inference relations

Some more properties

Left Logical Equivalence	if α and β are logically equivalent and $\alpha \mid\sim \gamma$, then $\beta \mid\sim \gamma$
Cautious Monotony	if $\alpha \mid\sim \beta$ and $\alpha \mid\sim \gamma$, then $\alpha \wedge \beta \mid\sim \gamma$

Characterization of preferential relations

Binary relation $\mid\sim$ is a preferential inference relation if and only if it satisfies *Left Logical Equivalence*, *Cautious Monotony*, *Right Weakening*, *Reflexivity*, *And* and *Or*

Kraus, Lehmann, Magidor, 1990

More nonmon inference relations

Rational inference relations

- Preferential plus

  Rational Monotony if $\alpha \wedge \beta \not\mid\!\sim \gamma$ and $\alpha \not\mid\!\sim \neg\beta$, then $\alpha \not\mid\!\sim \gamma$.
- Exactly inference relations defined by ranked preferential models

  Lehmann, Magidor 1992

Cumulative inference relations

- Arguably, the upper estimate to the class of nonmon inference relations

  Gabbay 1985, Makinson 1989
- Arguably, too broad

Nonmon logics for defining belief sets

Again focus on some models only

- Use theories of these models as candidate belief sets
- If more than one model, commit to one — anyone
- Now it is not about properties of nonmonotonic inference but about properties of belief sets (or models that define them)
- Sometimes easier to describe how to form belief sets than to characterize the corresponding class of models
- Typical constructions involve fixpoint conditions

Nonmon logics for defining belief sets

Most studied formalisms

- Default logic

  Reiter 1980
- Logic programming with stable-model semantics (more manageable fragment of default logic)

  Gelfond-Lifschitz, 1988
- Autoepistemic logic

  Moore 1984

Nonmon logics for defining belief sets

Multitude of different

- Intuitions
- Languages
- Constructions

And so key questions

- Are they connected?
- Are there any common abstract underlying principles?

Nonmon logics for defining belief sets

Multitude of different

- Intuitions
- Languages
- Constructions

And so key questions

- Are they connected?
- Are there any common abstract underlying principles?

For help — turn to algebra

Logic programming, default logic and autoepistemic logic

- Can be given a uniform algebraic treatment which
  - relates the semantics of these logics
  - suggests new semantics
  - highlights fundamental ideas behind these nonmon logics

Concepts, ideas, tools and approach

- Lattices and product lattices, operators and fixpoints
- Approximating mappings and operators, stable operators
- Knaster-Tarski Theorem
- Fitting's treatment of logic programming — generalized

For help — turn to algebra

Logic programming, default logic and autoepistemic logic

- Can be given a uniform algebraic treatment which
  - relates the semantics of these logics
  - suggests new semantics
  - highlights fundamental ideas behind these nonmon logics

Concepts, ideas, tools and approach

- Lattices and product lattices, operators and fixpoints
- Approximating mappings and operators, stable operators
- Knaster-Tarski Theorem
- Fitting's treatment of logic programming — generalized

Logic programming

Syntax: programs — collections of clauses

- $A \leftarrow B_1, \ldots, B_k, \mathbf{not}(C_1), \ldots, \mathbf{not}(C_m)$
- “if all B_i are derived and none of C_i can be, then derive A”

FOL semantics does not correspond to this reading

- $\{a \leftarrow not(b)\}$ has three models: $\{a\}$, $\{b\}$ and $\{a, b\}$
- only the first one “agrees” with the reading of the clause

Fundamental question: which semantics do?

- Supported and stable models (also 4-valued counterparts)
- Kripke-Kleene model, well-founded model (3-valued)

Logic programming

Syntax: programs — collections of clauses

- $A \leftarrow B_1, \ldots, B_k, \textbf{not}(C_1), \ldots, \textbf{not}(C_m)$
- "if all B_i are derived and none of C_i can be, then derive A"

FOL semantics does not correspond to this reading

- $\{a \leftarrow not(b)\}$ has three models: $\{a\}$, $\{b\}$ and $\{a, b\}$
- only the first one "agrees" with the reading of the clause

Fundamental question: which semantics do?

- Supported and stable models (also 4-valued counterparts)
- Kripke-Kleene model, well-founded model (3-valued)

Logic programming

Syntax: programs — collections of clauses

- $A \leftarrow B_1, \ldots, B_k, \mathbf{not}(C_1), \ldots, \mathbf{not}(C_m)$
- "if all B_i are derived and none of C_i can be, then derive A"

FOL semantics does not correspond to this reading

- $\{a \leftarrow not(b)\}$ has three models: $\{a\}$, $\{b\}$ and $\{a, b\}$
- only the first one "agrees" with the reading of the clause

Fundamental question: which semantics do?

- Supported and stable models (also 4-valued counterparts)
- Kripke-Kleene model, well-founded model (3-valued)

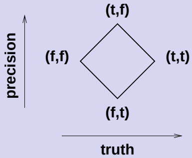
Logic programming algebraically (Fitting)

Lattice $\mathcal{TWO}$:

$$\begin{tikzcd} \boldsymbol{t} \arrow[d, dash] \\ \boldsymbol{f} \end{tikzcd}$$

- 2-valued interpretations assign elements of $\mathcal{TWO}$ to atoms
- Can be represented as sets (of true atoms)
- $I_{\mathcal{TWO}}$
- With inclusion $I_{\mathcal{TWO}}$ forms a complete lattice

Logic programming algebraically



Lattice $\mathcal{FOUR}$:

- 4-valued interpretations assign elements of $\mathcal{FOUR}$ to atoms
- $I_{\mathcal{FOUR}}$
- Can be represented as pairs of sets (I, J)
- Precision ordering
  - $(I, J) \leq_p (I', J')$ if $I \subseteq I'$ and $J' \subseteq J$
- Truth ordering
  - $(I, J) \leq_t (I', J')$ if $I \subseteq I'$ and $J \subseteq J'$
- With each ordering $I_{\mathcal{FOUR}}$ forms a complete lattice

Logic programming algebraically

Operators

- $A \leftarrow B_1, \dots, B_k, \mathbf{not}(C_1), \dots, \mathbf{not}(C_m)$
- $T_P(I) = \{head(r) \colon r \in P,\ \ I \models body(r)\}$
- T_P — operator on the lattice of 2-interpretations
- $\Psi_P(I, J) = \{head(r) \colon r \in P,\ \ I \models body^+(r), J \models body^-(r)\}$

  monotone wrt 1st arg; antimonotone wrt 2nd arg
- $\mathcal{T}_P(I, J) = (\Psi(I, J), \Psi(J, I))$

  monotone in $\langle I_{\mathcal{FOUR}}, \leq_p \rangle$
- $GL_P(I) = lfp(\Psi_P(\cdot, I))$

  antimonotone
- $\mathcal{GL}_P(I, J) = (GL_P(J), GL_P(I))$

  monotone in $\langle I_{\mathcal{FOUR}}, \leq_p \rangle$

Logic programming algebraically

Results

models of P	$\leftrightarrow$	prefixpoints of T_P
supported models of P	$\leftrightarrow$	fixpoints of T_P
stable models of P	$\leftrightarrow$	fixpoints of GL_P
partial supported models	$\leftrightarrow$	fixpoints of $\mathcal{T}_P$
KK model	$\leftrightarrow$	least fixpoint of $\mathcal{T}_P$
WFS model	$\leftrightarrow$	least fixpoint of $\mathcal{GL}_P$

Approximating mappings and operators

$\langle L, \leq \rangle$ — a complete lattice

- An *approximating mapping* — a mapping $A\colon L^2 \to L$ such that for every $x \in L$, the operator $A(\cdot, x)$ is monotone and the operator $A(x, \cdot)$ is antimonotone
- An *approximating operator*:
  $\mathcal{A}(I, J) = (A(I, J), A(J, I))$
- Approximating operators are monotone with respect to precision ordering on $\langle L^2, \leq_p \rangle$
  $(x, y) \leq_p (x', y')$ if $x \leq x'$ and $y' \leq y$
- If O is an operator on L such that $O(x) = A(x, x)$, then A and $\mathcal{A}$ are an *approximating mapping* and *approximating operator for O*, respectively

Approximating mappings and operators

Intuitions

- If $x, y, z \in L$ and $x \leq z \leq y$, then (x, y) is an *approximation* of z
- If A is an approximating mapping for O and (x, y) is an approximation to z then

$$A(x,y) \leq A(x,z) \leq A(z,z) = O(z) \leq A(z,x) \leq A(y,x)$$

- Consequently $(A(x,y), A(y,x))$ approximates $O(z)$.
- Or — $\mathcal{A}(x,y)$ approximates $O(z)$.

Approximating mappings and operators

Existence

- Every operator O has an approximating mapping:

$$A_O(x, y) = \begin{cases} \perp & \text{if } x < y \\ O(x) & \text{if } x = y \\ \top & \text{otherwise.} \end{cases}$$

- Every operator O has an approximating operator:

$$\mathcal{A}_O(x, y) = (A_O(x, y), A_O(y, x))$$

- Approximating mappings and operators are not unique (in general)

Approximating mappings and operators

Special cases

- If O is monotone:
  $C_O(x,y) = O(x)$, for $x, y \in L$
  $\mathcal{C}_O(x,y) = (O(x), O(y))$, for $x, y \in L$
- If O is antimonotone:
  $C_O(x,y) = O(y)$, for $x, y \in L$
  $\mathcal{C}_O(x,y) = (O(y), O(x))$, for $x, y \in L$
- In each case:
  - C_O is an approximating mapping for O
  - $\mathcal{C}_O$ is an approximating operator for O
- *Canonical* approximating mapping (operator)

Stable operators

O — an operator on L
A — an approximating mapping for O

- An *A-stable operator* for O on L is an operator S_A on L such that for every $y \in L$:
$$S_A(y) = \mathit{lfp}(A(\cdot, y))$$
- An *A-stable operator* for O on L^2 is an operator $\mathcal{S}_A$ on L^2 such that for every $y \in L$:
$$\mathcal{S}_A(x, y) = (S_A(y), S_A(x))$$
- $\mathcal{S}_A$ is an approximating operator for S_A

Stable fixpoints

O — an operator on L
A — an approximating mapping for O

- An element $(x, y) \in L^2$ is a *general A-stable fixpoint* of O if $(x, y) = \mathcal{S}_A(x, y)$
- An element $x \in L$ is an *A-stable fixpoint* of O if $x = S_A(x)$
  if and only if $(x, x) = \mathcal{S}_A(x, x)$
- $St(O, A)$ — the set of A-stable fixpoints of O

Back to LP for a moment

What's what?

$$\begin{array}{ccc} O & \leftrightarrow & T_P \\ A & \leftrightarrow & \Psi_P \\ S_A & \leftrightarrow & GL_P \end{array}$$

- Only now we do not have a single fixed approximating mapping

Properties

O — an operator on L
A — an approximating mapping for O

- S_A is antimonotone

  in particular: $\mathcal{S}_A$ is the canonical approximating operator for S_A
- $\mathcal{S}_A$ is $\leq_p$-monotone and $\leq_t$-antimonotone
- Fixpoints of $\mathcal{S}_A$ are $\leq_t$-minimal fixpoints of A
- Complete fixpoints of $\mathcal{S}_A$ correspond to fixpoints of S_A
- Complete fixpoints of $\mathcal{S}_A$ are fixpoints of O
- $\leq_p$-least fixpoint of $\mathcal{S}_A$ — well-founded fixpoint of A
- KK fixpoint of $A \leq_p$ WF fixpoint of A
- All these concepts and results specialize to known concepts and results in logic programming

Properties

Ultimate semantics

- How to choose approximating mappings?
- In LP, DL, AEL they pop up naturally — but in general?
- The precision ordering extends to approximating operators
- Every operator has a most precise, ultimate, approximating operator
- It defines:
  - a class of ultimate stable fixpoints
  - the ultimate KK fixpoint (at least as precise as all other KK fixpoints)
  - the ultimate WF fixpoint (at least as precise as all other WF fixpoints)
- For LP, DL, AEL — different than standard semantics but with several nice properties

Does the approach apply to default logic?

Default

- $d = \frac{\alpha : \beta_1, \ldots, \beta_k}{\gamma}$
  - α — the prerequisite
  - β_i, $1 \leq i \leq k$ — the justifications
  - γ — the consequent
- Inference rule with the following informal reading:
  conclude γ if α holds and if all justifications β_i are possible

Example

- $\frac{prof(X) : teaches(X)}{teaches(X)}$

Does the approach apply to default logic?

Default

- $d = \frac{\alpha : \beta_1, \ldots, \beta_k}{\gamma}$
  - α — the prerequisite
  - β_i, $1 \leq i \leq k$ — the justifications
  - γ — the consequent
- Inference rule with the following informal reading:
  conclude γ if α holds and if all justifications β_i are possible

Example

- $\frac{prof(X) : teaches(X)}{teaches(X)}$

Default logic

Default theory

- Default theory — a pair (D, W), where
  - W is a set of formulas
  - D is a set of defaults
- W represents our knowledge, in general, incomplete
- Defaults in D serve as "meta-rules" we use to fill in gaps in what we know

Default logic

Extensions (propositional case)

- (D, W) — a default theory
- S — a belief set (ie, a theory closed under consequence); $W \subseteq S$
- $\Delta = (D, W)$ "revises" S
- $\Gamma_\Delta(S)$ is the least set U such that:
  - U is closed under propositional provability
  - $W \subseteq U$
  - for every default $d \in D$,
    if $p(d) \in U$ and for every $\beta \in j(d)$, $S \not\vdash \neg\beta$, then $c(d) \in U$.
- Fixpoints of Γ_Δ represent belief sets consistent with W that are in a way stable with respect to Δ — they cannot be revised away
- Reiter defined extensions of (D, W) as fixpoints of $\Gamma_\Delta(S)$

Back to the university-professor scenario

$\frac{prof_J : teaches_J}{teaches_J}$

- $W = \{prof_J, chair_J \supset \neg teaches_J\}$
- One extension: $Cn(W \cup \{teaches_J\})$

- $W = \{prof_J, chair_J \supset \neg teaches_J, chair_J\}$
- One extension: $Cn(W \cup \{\neg teaches_J\})$

Back to the university-professor scenario

$$\frac{prof_J : teaches_J}{teaches_J}$$

- $W = \{prof_J, chair_J \supset \neg teaches_J\}$
- One extension: $Cn(W \cup \{teaches_J\})$

- $W = \{prof_J, chair_J \supset \neg teaches_J, chair_J\}$
- One extension: $Cn(W \cup \{\neg teaches_J\})$

Default logic

Algebraic perspective: need a lattice and operators

- Lattice of all sets closed under consequence and containing W
- Default $d = \frac{\alpha:\, \beta_1,\ldots,\beta_k}{\gamma}$ is (S, S')-applicable if
  - $S \vdash \alpha$
  - $S' \not\vdash \neg\beta_i$
- Basic operator:

$$E_\Delta(S) = Cn(\{cons(d) \colon d \in D,\ \ d \text{ is } (S,S)\text{-applicable}\})$$

- Basic approximating mapping:

$$A_{E_\Delta}(S, S') = Cn(\{cons(d) \colon d \in D,\ \ d \text{ is } (S,S')\text{-applicable}\})$$

- Γ_Δ is an A_{E_Δ}-stable operator for E_Δ

So what do we get?

Default logic

- Default logic — an instance of the algebraic theory of approximating mappings and operators
- General results specialize to well known properties of default theories (antichain property, splitting results)
- Ultimate semantics — new

Beyond default logic

- Since AEL can be given the same treatment — a unified view of default and autoepistemic logics
- An abstract perspective on the concept of equivalence of nonmon theories

So what do we get?

Default logic

- Default logic — an instance of the algebraic theory of approximating mappings and operators
- General results specialize to well known properties of default theories (antichain property, splitting results)
- Ultimate semantics — new

Beyond default logic

- Since AEL can be given the same treatment — a unified view of default and autoepistemic logics
- An abstract perspective on the concept of equivalence of nonmon theories

Motivation

Knowledge base rewriting

- Knowledge base - a collection of interrelated modules
- $KB_1 \cup KB_2$
- Knowledge base rewriting: replace one module, say KB_1, with another, say KB_1', without changing the meaning of the knowledge base
- When are two modules equivalent for replacement?
  - If $KB_1 \cup KB_2$ and $KB_1' \cup KB_2$ have the same meaning
    not quite what we want - depends on KB_2
  - If $KB_1 \cup KB$ and $KB_1' \cup KB$ have the same meaning for every knowledge base KB
    better

Equivalence for replacement

Classical logic

- *KB* modules — FOL theories
- The meaning specified by the standard FOL semantics
- Logical equivalence is necessary and sufficient condition for the equivalence for replacement

Nonmon logics

Not quite as straightforward

Equivalence for replacement

Classical logic

- *KB* modules — FOL theories
- The meaning specified by the standard FOL semantics
- Logical equivalence is necessary and sufficient condition for the equivalence for replacement

Nonmon logics

Not quite as straightforward

Logic programming

The meaning is given by stable models

- Equivalence for replacement — for every program R, programs $P \cup R$ and $Q \cup R$ have the same stable models
- Known as strong equivalence

  Lifschitz, Pearce, Valverde; Lin; Turner; Eiter, Fink, Woltran
- Different than logical equivalence
  - $\{p \leftarrow \textbf{not}(q)\}$ and $\{q \leftarrow \textbf{not}(p)\}$
  - The same models but different meaning
- Different than "nonmonotonic" equivalence
  - $P = \{p\}$ and $Q = \{p \leftarrow \textbf{not}(q)\}$
  - The same stable models ($\{p\}$)
  - But, $P \cup \{q\}$ and $Q \cup \{q\}$ have different stable models! ($\{p, q\}$ and $\{q\}$, respectively)

When are two programs strongly equivalent?

Se-model characterization

- A pair (X, Y) of sets of atoms is an se-model of a program P if
  - $X \subseteq Y$
  - $T_P(Y) \subseteq Y \quad \rightarrow \quad Y$ is a model of P
  - $\Psi_P(X, Y) \subseteq X \quad \rightarrow \quad X$ is a model of P^Y
- Logic programs P and Q are strongly equivalent iff they have the same se-models
- A similar concept characterizes strong equivalence of default theories (Turner)
- Once more, algebra provides a more general abstract perspective

Strong equivalence of operators

Extending lattice operators

- P and R — operators on L
- An *extension* of P with R — an operator $P \vee R$

$$(P \vee R)(x) = P(x) \vee R(x),$$

  for every $x \in L$
- R — an *extending* operator
- Back to LP: if P and R are programs, then $T_{P \cup R} = T_P \vee T_R$

Strong equivalence of operators

Key question: which stable fixpoints to consider?

- Operators P and Q must come with approximating mappings
- Extending operators R, too!
- Which approximating mappings to use for $P \vee R$ and $Q \vee R$?
- $A_P \vee A_R$ and $A_Q \vee A_R$, respectively!

Strong equivalence of operators

Definition

- P and Q — operators on L
- A_P and A_Q — their approximating mappings, respectively
- P and Q are *strongly equivalent* with respect to (A_P, A_Q) if for every operator R and every approximating mapping A_R of R,

$$St(P \vee R, A_P \vee A_R) = St(Q \vee R, A_Q \vee A_R).$$

- $P \equiv_s Q$ w/r to (A_P, A_Q)

Problem

- When are two operators, P and Q, strongly equivalent with respect to (A_P, A_Q)?
(where A_P and A_Q are approximating mappings for P and Q)

Strong equivalence of operators

Definition

- P and Q — operators on L
- A_P and A_Q — their approximating mappings, respectively
- P and Q are *strongly equivalent* with respect to (A_P, A_Q) if for every operator R and every approximating mapping A_R of R,

$$St(P \vee R, A_P \vee A_R) = St(Q \vee R, A_Q \vee A_R).$$

- $P \equiv_s Q$ w/r to (A_P, A_Q)

Problem

- When are two operators, P and Q, strongly equivalent with respect to (A_P, A_Q)?
(where A_P and A_Q are approximating mappings for P and Q)

Se-pairs

Definition

- P — an operator on L
- A_P — an approximating mapping for P
- A pair $(x, y) \in L^2$ is an *se-pair* for P w/r to A_P if:

$$\begin{array}{ll} \textbf{SE1:} & x \leq y \\ \textbf{SE2:} & P(y) \leq y \\ \textbf{SE3:} & A_P(x, y) \leq x \end{array}$$

- $SE(P, A_P)$ — the set of all se-pairs for P w/r to A_P
- Generalize se-models by Turner

Characterizing strong equivalence

Theorem

- P and Q — operators on a complete lattice L
- A_P and A_Q — approximating mappings for P and Q, respectively
- If $SE(P, A_P) = SE(Q, A_Q)$ then $P \equiv_s Q$ w/r to (A_P, A_Q)
- That is, for every operator R and every approximating mapping A_R for R, $St(P \vee R, A_P \vee A_R) = St(Q \vee R, A_Q \vee A_R)$

Characterizing strong equivalence

Converse theorem

- It holds
- But a stronger result holds, too!

Characterizing strong equivalence

Converse theorem

- It holds
- But a stronger result holds, too!

Characterizing strong equivalence

Simple operators

- An operator R is *simple* if for some $x, y \in L$ such that $x \leq y$, we have
$$R(z) = \begin{cases} y & \text{if } x < z \\ x & \text{otherwise} \end{cases}$$
for every $z \in L$.
- Constant operators are simple (take $x = y =$ the single value of the operator)
- Simple operators are monotone
- If for every simple operator R, $St(P \vee R, A_P \vee C_R) = St(Q \vee R, A_Q \vee C_R)$ then $SE(P, A_P) = SE(Q, A_Q)$.

Characterizing strong equivalence

Theorem

- $P \equiv_s Q$ w/r to (A_P, A_Q) if and only if $SE(P, A_P) = SE(Q, A_Q)$
- Perhaps more interestingly ...
- for every operator R and for every approximating mapping A_R for R, $St(P \vee R, A_P \vee A_R) = St(Q \vee R, A_Q \vee A_R)$ ($P \equiv_s Q$)
  iff
  for every simple operator R,
  $St(P \vee R, A_P \vee C_R) = St(Q \vee R, A_Q \vee C_R)$

Characterizing strong equivalence

Theorem

- $P \equiv_s Q$ w/r to (A_P, A_Q) if and only if $SE(P, A_P) = SE(Q, A_Q)$
- Perhaps more interestingly ...
- for every operator R and for every approximating mapping A_R for R, $St(P \vee R, A_P \vee A_R) = St(Q \vee R, A_Q \vee A_R)$ ($P \equiv_s Q$)
  iff
  for every simple operator R,
  $St(P \vee R, A_P \vee C_R) = St(Q \vee R, A_Q \vee C_R)$

Recap and future

What did we do?

- Outlined major trends in nonmon logic research
- Discussed in more detail algebraic foundations of nonmon logics defining belief sets
- Demonstrated usefulness of the algebraic approach

Much left out: uniform equivalence, properties of ultimate semantics, splitting theorems

Many questions, here just one example

- Each set D of defaults defines an inference relation:
  $\alpha \mid\!\sim_D \beta$ if β is in every extension of $(D, \{\alpha\})$
- These relations are not cumulative, preferential nor rational
- Can cumulative (preferential, rational) inference relations be characterized in terms of some fixpoint semantics for DL?

Recap and future

What did we do?

- Outlined major trends in nonmon logic research
- Discussed in more detail algebraic foundations of nonmon logics defining belief sets
- Demonstrated usefulness of the algebraic approach

Much left out: uniform equivalence, properties of ultimate semantics, splitting theorems

Many questions, here just one example

- Each set D of defaults defines an inference relation:
  $\alpha \mid\!\sim_D \beta$ if β is in every extension of $(D, \{\alpha\})$
- These relations are not cumulative, preferential nor rational
- Can cumulative (preferential, rational) inference relations be characterized in terms of some fixpoint semantics for DL?

Thank you!

Acknowledgments

- Marc Denecker and Victor Marek
- Gerd Brewka and Ilkka Niemelä