Strong Equivalence and Relatives — Logically and Algebraically

Mirosław Truszczyński

Department of Computer Science
University of Kentucky

September 8, 2007

Motivation (1)

Query optimization

- Compute answers to a query Q from a knowledge base KB
  reason from $Q \cup KB$
- Rewrite Q into an equivalent query Q', which can be processed more efficiently
  reasoning from $Q' \cup KB$ *easier*
- When are two queries equivalent?
  - If $Q \cup KB$ and $Q' \cup KB$ have the same meaning
    not quite what we want — knowledge-base dependent
  - If $Q \cup KB$ and $Q' \cup KB$ have the same meaning for every knowledge base KB
    better — knowledge-base independent

Motivation (2)

Knowledge base rewriting

- Knowledge base — a collection of interrelated modules (say, answer-set programs)
- Knowledge base rewriting: replace one module with another without changing the meaning of the knowledge base
- When are two modules equivalent for replacement?
  - The same two basic options as above

In each scenario, it is the second option that we are after

Motivation (2)

Knowledge base rewriting

- Knowledge base — a collection of interrelated modules (say, answer-set programs)
- Knowledge base rewriting: replace one module with another without changing the meaning of the knowledge base
- When are two modules equivalent for replacement?
  - The same two basic options as above

In each scenario, it is the second option that we are after

Equivalence for replacement (1)

Classical logic

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

Equivalence for replacement (2)

Logic programming

- The meaning is given by stable models (answer sets)
- Equivalence for substitution — for every program R, programs $P \cup R$ and $Q \cup R$ have the same stable models
- Known as strong equivalence

  Lifschitz, Pearce, Valverde 2001; Lin 2002; Turner 2003; Eiter, Fink 2003; Eiter, Fink, Tompits, Woltran, 2005
- Different than logical equivalence
  - $\{p \leftarrow \mathbf{not}(q)\}$ and $\{q \leftarrow \mathbf{not}(p)\}$
  - The same models but different meaning
- Different than nonmonotonic equivalence
  - $P = \{p\}$ and $Q = \{p \leftarrow \mathbf{not}(q)\}$
  - The same stable models; $\{p\}$ is the only stable model in each case
  - 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
  - $X \subseteq Y$
  - $Y \models P$
  - $X \models P^Y$
- 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 2003

What's behind strong equivalence?

Logics (albeit non-standard)

- Logic here-and-there

  Lifschitz, Pearce, Valverde, 2001; Lifschitz, Ferraris, 2005

- Modal logics S4F and SW5

  Cabalar 2004, MT 2007

Algebra

- Lattices, operators and fixpoints

  MT 2006

What's behind strong equivalence?

Logics (albeit non-standard)

- Logic here-and-there

  Lifschitz, Pearce, Valverde, 2001; Lifschitz, Ferraris, 2005

- Modal logics S4F and SW5

  Cabalar 2004, MT 2007

Algebra

- Lattices, operators and fixpoints

  MT 2006

Logic here-and-there, Heyting 1930

Syntax

- Connectives: $\bot$, $\vee$, $\wedge$, $\to$
- Formulas - standard extension of atoms by means of connectives
- $\neg\varphi$ - shorthand for $\varphi \to \bot$
- Language $\mathcal{L}_{ht}$

Why important?

- Disjunctive logic programs — special theories in $\mathcal{L}_{ht}$
  change the direction of implication
- General logic programs (Ferraris, Lifschitz) $=$ theories in $\mathcal{L}_{ht}$
  answer-set semantics extends to general logic programs and so to theories in $\mathcal{L}_{ht}$

Logic here-and-there, Heyting 1930

Syntax

- Connectives: $\bot$, $\vee$, $\wedge$, $\rightarrow$
- Formulas - standard extension of atoms by means of connectives
- $\neg\varphi$ - shorthand for $\varphi \rightarrow \bot$
- Language $\mathcal{L}_{ht}$

Why important?

- Disjunctive logic programs — special theories in $\mathcal{L}_{ht}$
  change the direction of implication
- General logic programs (Ferraris, Lifschitz) $=$ theories in $\mathcal{L}_{ht}$
  answer-set semantics extends to general logic programs and so to theories in $\mathcal{L}_{ht}$

Entailment in logic here-and-there

Ht-interpretations

- Pairs $\langle H, T \rangle$, where $H \subseteq T$ are sets of atoms
- Kripke interpretations with two worlds “here” and “there”
  - H determines the valuation for “here”
  - T determines the valuation for “there”

Kripke-model satisfiability in the world “here” $\models_{ht}$

- $\langle H, T \rangle \not\models_{ht} \bot$
- $\langle H, T \rangle \models_{ht} p$ if $p \in H$ (for atoms only)
- $\langle H, T \rangle \models_{ht} \varphi \wedge \psi$ and $\langle H, T \rangle \models_{ht} \varphi \vee \psi$ — standard recursion
- $\langle H, T \rangle \models_{ht} \varphi \rightarrow \psi$ if
  - $\langle H, T \rangle \not\models_{ht} \varphi$ or $\langle H, T \rangle \models_{ht} \psi$
  - $T \models \varphi \rightarrow \psi$ (in standard propositional logic).
- If $\langle H, T \rangle \models_{ht} \varphi$ $\langle H, T \rangle$ an ht-model of φ
- φ and ψ are ht-equivalent if they have the same ht-models

Entailment in logic here-and-there

Ht-interpretations

- Pairs $\langle H, T \rangle$, where $H \subseteq T$ are sets of atoms
- Kripke interpretations with two worlds "here" and "there"
  - H determines the valuation for "here"
  - T determines the valuation for "there"

Kripke-model satisfiability in the world "here" $\models_{ht}$

- $\langle H, T \rangle \not\models_{ht} \bot$
- $\langle H, T \rangle \models_{ht} p$ if $p \in H$ (for atoms only)
- $\langle H, T \rangle \models_{ht} \varphi \wedge \psi$ and $\langle H, T \rangle \models_{ht} \varphi \vee \psi$ — standard recursion
- $\langle H, T \rangle \models_{ht} \varphi \rightarrow \psi$ if
  - $\langle H, T \rangle \not\models_{ht} \varphi$ or $\langle H, T \rangle \models_{ht} \psi$
  - $T \models \varphi \rightarrow \psi$ (in standard propositional logic).
- If $\langle H, T \rangle \models_{ht} \varphi$ $\langle H, T \rangle$ an ht-model of φ
- φ and ψ are ht-equivalent if they have the same ht-models

Proof theory

Natural deduction — sequents and rules

- Sequents $\Gamma \Rightarrow \varphi$ — "F under the assumptions Γ"
- Introduction rules for $\wedge$, $\vee$, $\rightarrow$

$$\frac{\Gamma \Rightarrow \varphi \quad \Delta \Rightarrow \psi}{\Gamma, \Delta \Rightarrow \varphi \wedge \psi}$$

- Elimination rules for $\wedge$, $\vee$, $\rightarrow$

$$\frac{\Gamma \Rightarrow \varphi \quad \Delta \Rightarrow \varphi \rightarrow \psi}{\Gamma, \Delta \Rightarrow \psi}$$

- Contradiction

$$\frac{\Gamma \Rightarrow \bot}{\Gamma \Rightarrow \varphi}$$

- Weakening

$$\frac{\Gamma \Rightarrow \varphi}{\Gamma' \Rightarrow \varphi} \qquad \text{for all } \Gamma', \Gamma \text{ s.t. } \Gamma' \subseteq \Gamma$$

Proof theory (2)

Axiom schemas

(AS1)	$\varphi \Rightarrow \varphi$	
(AS2)	$\Rightarrow \varphi \vee \neg\varphi$	(Excluded Middle)
(AS2′)	$\Rightarrow \neg\varphi \vee \neg\neg\varphi$	(Weak EM)
(AS2″)	$\Rightarrow \varphi \wedge (\varphi \rightarrow \psi) \wedge \neg\psi$	(in between (AS2) and (AS2′))

Logics through natural deduction

Propositional logic	(AS1), (AS2)
Intuitionistic logic	(AS1)
Logic here-and-there	(AS1),(AS2″)

In particular

- φ and ψ are ht-equivalent iff $\Rightarrow \varphi \leftrightarrow \psi$ has a proof from (AS1) and (AS2″)

Proof theory (2)

Axiom schemas

(AS1)	$\varphi \Rightarrow \varphi$	
(AS2)	$\Rightarrow \varphi \vee \neg\varphi$	(Excluded Middle)
(AS2′)	$\Rightarrow \neg\varphi \vee \neg\neg\varphi$	(Weak EM)
(AS2″)	$\Rightarrow \varphi \wedge (\varphi \rightarrow \psi) \wedge \neg\psi$	(in between (AS2) and (AS2′))

Logics through natural deduction

Propositional logic	(AS1), (AS2)
Intuitionistic logic	(AS1)
Logic here-and-there	(AS1),(AS2″)

In particular

- φ and ψ are ht-equivalent iff $\Rightarrow \varphi \leftrightarrow \psi$ has a proof from (AS1) and (AS2″)

Proof theory (2)

Axiom schemas

(AS1)	$\varphi \Rightarrow \varphi$	
(AS2)	$\Rightarrow \varphi \vee \neg\varphi$	(Excluded Middle)
(AS2′)	$\Rightarrow \neg\varphi \vee \neg\neg\varphi$	(Weak EM)
(AS2″)	$\Rightarrow \varphi \wedge (\varphi \rightarrow \psi) \wedge \neg\psi$	(in between (AS2) and (AS2′))

Logics through natural deduction

Propositional logic	(AS1), (AS2)
Intuitionistic logic	(AS1)
Logic here-and-there	(AS1),(AS2″)

In particular

- φ and ψ are ht-equivalent iff $\Rightarrow \varphi \leftrightarrow \psi$ has a proof from (AS1) and (AS2″)

Logic here-and-there and ASP

Equilibrium models, Pearce 1997

- $\langle T, T \rangle$ is an *equilibrium model* of a set A of formulas if
  - $\langle T, T \rangle \models_{ht} A$, and
  - for every $H \subseteq T$ such that $\langle H, T \rangle \models_{ht} A$, $H = T$

Key connection

- A set M of atoms is an answer set of a disjunctive logic program P (general logic program P) if and only if $\langle M, M \rangle$ is an equlibrium model for P

Strong equivalence

- Let P and Q be two (general) programs. The following conditions are equivalent:
  - P and Q are strongly equivalent
  - P and Q are ht-equivalent
  - P and Q have the same ht-models
  - $\Rightarrow P \leftrightarrow Q$ has a proof from (AS1) and (AS2$''$)

Logic here-and-there and ASP

Equilibrium models, Pearce 1997

- $\langle T, T \rangle$ is an *equilibrium model* of a set A of formulas if
  - $\langle T, T \rangle \models_{ht} A$, and
  - for every $H \subseteq T$ such that $\langle H, T \rangle \models_{ht} A$, $H = T$

Key connection

- A set M of atoms is an answer set of a disjunctive logic program P (general logic program P) if and only if $\langle M, M \rangle$ is an equlibrium model for P

Strong equivalence

- Let P and Q be two (general) programs. The following conditions are equivalent:
  - P and Q are strongly equivalent
  - P and Q are ht-equivalent
  - P and Q have the same ht-models
  - $\Rightarrow P \leftrightarrow Q$ has a proof from (AS1) and (AS2$''$)

Logic here-and-there and ASP

Equilibrium models, Pearce 1997

- $\langle T, T\rangle$ is an *equilibrium model* of a set A of formulas if
  - $\langle T, T\rangle \models_{ht} A$, and
  - for every $H \subseteq T$ such that $\langle H, T\rangle \models_{ht} A$, $H = T$

Key connection

- A set M of atoms is an answer set of a disjunctive logic program P (general logic program P) if and only if $\langle M, M\rangle$ is an equlibrium model for P

Strong equivalence

- Let P and Q be two (general) programs. The following conditions are equivalent:
  - P and Q are strongly equivalent
  - P and Q are ht-equivalent
  - P and Q have the same ht-models
  - $\Rightarrow P \leftrightarrow Q$ has a proof from (AS1) and (AS2$''$)

Modal logics

The language $\mathcal{L}_K$

- $\varphi ::= \bot \mid p \mid K\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$ (where p - an atom)
  e.g.: $a \rightarrow K(\neg b \wedge K(a \vee \neg b))$

Proof theory

- Modus ponens and necesitation $\frac{\varphi}{K\varphi}$
- Modal axioms such as:
  - **K:** $K(\varphi \rightarrow \psi) \rightarrow (K\varphi \rightarrow K\psi)$
  - **T:** $K\varphi \rightarrow \varphi$
  - **4:** $K\varphi \rightarrow KK\varphi$
  - **F:** $(\varphi \wedge \neg K\neg K\psi) \rightarrow K(\neg\varphi \vee \psi)$
  - **5:** $\neg K\neg K\varphi \rightarrow K\varphi$
- Logics determined by modal axioms
  - Modal logic S4F: K, T, 4, F
  - Modal logic S5: K, T, 4, 5

Modal logics

The language $\mathcal{L}_K$

- $\varphi ::= \bot \mid p \mid K\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$ (where p - an atom)
  e.g.: $a \rightarrow K(\neg b \wedge K(a \vee \neg b))$

Proof theory

- Modus ponens and necesitation $\frac{\varphi}{K\varphi}$
- Modal axioms such as:
  - **K:** $K(\varphi \rightarrow \psi) \rightarrow (K\varphi \rightarrow K\psi)$
  - **T:** $K\varphi \rightarrow \varphi$
  - **4:** $K\varphi \rightarrow KK\varphi$
  - **F:** $(\varphi \wedge \neg K \neg K \psi) \rightarrow K(\neg\varphi \vee \psi)$
  - **5:** $\neg K \neg K \varphi \rightarrow K\varphi$
- Logics determined by modal axioms
  - Modal logic S4F: K, T, 4, F
  - Modal logic S5: K, T, 4, 5

Modal logics

The language $\mathcal{L}_K$

- $\varphi ::= \bot \mid p \mid K\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$ (where p - an atom)
  e.g.: $a \rightarrow K(\neg b \wedge K(a \vee \neg b))$

Proof theory

- Modus ponens and necesitation $\frac{\varphi}{K\varphi}$
- Modal axioms such as:
  - **K:** $K(\varphi \rightarrow \psi) \rightarrow (K\varphi \rightarrow K\psi)$
  - **T:** $K\varphi \rightarrow \varphi$
  - **4:** $K\varphi \rightarrow KK\varphi$
  - **F:** $(\varphi \wedge \neg K\neg K\psi) \rightarrow K(\neg\varphi \vee \psi)$
  - **5:** $\neg K\neg K\varphi \rightarrow K\varphi$
- Logics determined by modal axioms
  - Modal logic S4F: K, T, 4, F
  - Modal logic S5: K, T, 4, 5

Modal logics (2)

Kripke semantics

- $\langle W, A, \pi \rangle$
- Classes of Kripke models characterize modal logics
- Logic S5
  - models with universal accessibility relation $\langle W, \pi \rangle$
- Logic S4F
  - S4F-interpretations: $\langle V, W, \pi \rangle$
  - $\mathcal{M}, w \models \varphi$ ($w \in V \cup W$ and $\varphi \in \mathcal{L}_K$)
    - ⋆ $\mathcal{M}, w \not\models \bot$
    - ⋆ $\mathcal{M}, w \models p$ if $p \in \pi(w)$ (for $p \in At$)
    - ⋆ If $w \in V$, then $\mathcal{M}, w \models K\varphi$ if $\mathcal{M}, v \models \varphi$ for every $v \in V \cup W$
    - ⋆ If $w \in W$, then $\mathcal{M}, w \models K\varphi$ if $\mathcal{M}, v \models \varphi$ for every $v \in W$
    - ⋆ The induction over boolean connectives is standard
  - $\mathcal{M} \models \varphi$ if $\mathcal{M}, w \models \varphi$, for every $w \in V \cup W$; S4F-models

Modal logics (2)

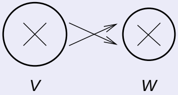
Kripke semantics

- $\langle W, A, \pi \rangle$
- Classes of Kripke models characterize modal logics
- Logic S5
  - models with universal accessibility relation $\langle W, \pi \rangle$
- Logic S4F
  - S4F-interpretations: $\langle V, W, \pi \rangle$
  - $\mathcal{M}, w \models \varphi$ ($w \in V \cup W$ and $\varphi \in \mathcal{L}_K$)
    - $\mathcal{M}, w \not\models \bot$
    - $\mathcal{M}, w \models p$ if $p \in \pi(w)$ (for $p \in At$)
    - If $w \in V$, then $\mathcal{M}, w \models K\varphi$ if $\mathcal{M}, v \models \varphi$ for every $v \in V \cup W$
    - If $w \in W$, then $\mathcal{M}, w \models K\varphi$ if $\mathcal{M}, v \models \varphi$ for every $v \in W$
    - The induction over boolean connectives is standard
  - $\mathcal{M} \models \varphi$ if $\mathcal{M}, w \models \varphi$, for every $w \in V \cup W$; S4F-models

Modal logics (2)

Kripke semantics

- $\langle W, A, \pi \rangle$
- Classes of Kripke models characterize modal logics
- Logic S5
  - models with universal accessibility relation $\langle W, \pi \rangle$
- Logic S4F
  - S4F-interpretations: $\langle V, W, \pi \rangle$



  - $\mathcal{M}, w \models \varphi$ ($w \in V \cup W$ and $\varphi \in \mathcal{L}_K$)
    - ⋆ $\mathcal{M}, w \not\models \bot$
    - ⋆ $\mathcal{M}, w \models p$ if $p \in \pi(w)$ (for $p \in At$)
    - ⋆ If $w \in V$, then $\mathcal{M}, w \models K\varphi$ if $\mathcal{M}, v \models \varphi$ for every $v \in V \cup W$
    - ⋆ If $w \in W$, then $\mathcal{M}, w \models K\varphi$ if $\mathcal{M}, v \models \varphi$ for every $v \in W$
    - ⋆ The induction over boolean connectives is standard
  - $\mathcal{M} \models \varphi$ if $\mathcal{M}, w \models \varphi$, for every $w \in V \cup W$; S4F-models

Modal nonmonotonic logics

Expansions

- $\mathcal{S}$ — modal (monotone) logic; $\models_{\mathcal{S}}$
- $\mathcal{S}$-expansion of a modal theory $I \subseteq \mathcal{L}_K$:

$$T = \{\varphi \in \mathcal{L}_K | I \cup \{\neg K\varphi | \varphi \in \mathcal{L}_K \setminus T\} \models_{\mathcal{S}} \varphi\},$$

Nonmonotonic S4F captures (T_, 1991; Schwarz and T_, 1994)

- (Disjunctive) logic programming with the answer set semantics
- (Disjunctive) default logic
- General default logic *(Cabalar, 2004; extended by T_, 2007)*
- Logic of grounded knowledge
- Logic of minimal belief and negation as failure
- Logic of minimal knowledge and belief
- Is S4F the logic underlying nonmon reasonig?

Modal nonmonotonic logics

Expansions

- $\mathcal{S}$ — modal (monotone) logic; $\models_{\mathcal{S}}$
- $\mathcal{S}$-expansion of a modal theory $I \subseteq \mathcal{L}_K$:

$$T = \{\varphi \in \mathcal{L}_K | I \cup \{\neg K\varphi | \varphi \in \mathcal{L}_K \setminus T\} \models_{\mathcal{S}} \varphi\},$$

Nonmonotonic S4F captures (T_, 1991; Schwarz and T_, 1994)

- (Disjunctive) logic programming with the answer set semantics
- (Disjunctive) default logic
- General default logic *(Cabalar, 2004; extended by T_, 2007)*
- Logic of grounded knowledge
- Logic of minimal belief and negation as failure
- Logic of minimal knowledge and belief
- Is S4F the logic underlying nonmon reasonig?

Is S4F the logic underlying nonmon reasoning?

I claim: yes!

- But some restrictions on the language are needed
- If $I, J \subseteq \mathcal{L}_K$ have the same S4F-models then for every $K \subseteq \mathcal{L}_K$, $I \cup T$ and $J \cup T$ have the same S4F-expansions
- The converse does not hold!

Modal defaults and modal default theories

- $\varphi ::= K\psi \mid K\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$
  where ψ — a propositional formula
- For modal default theories (sets of modal defaults) S4F characterizes strong equivalence!

Is S4F the logic underlying nonmon reasoning?

I claim: yes!

- But some restrictions on the language are needed
- If $I, J \subseteq \mathcal{L}_K$ have the same S4F-models then for every $K \subseteq \mathcal{L}_K$, $I \cup T$ and $J \cup T$ have the same S4F-expansions
- The converse does not hold!

Modal defaults and modal default theories

- $\varphi ::= K\psi \mid K\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi$
  where ψ — a propositional formula
- For modal default theories (sets of modal defaults) S4F characterizes strong equivalence!

First, the semantics simplifies!

Se-pairs

- $\langle L, U \rangle$ - L, U are propositional theories closed under propositional entailment
- Entailment relations $\models_u$ and $\models_l$ for modal defaults
- $\langle L, U \rangle \models_u \varphi$
  - $\varphi = K\psi$, where ψ is propositional
    $\langle L, U \rangle \models_u \varphi$ if $\psi \in U$
  - Boolean connectives standard
  - $\varphi = K\psi$, where ψ is a modal default
    $\langle L, U \rangle \models_u \varphi$ if $\langle L, U \rangle \models_u \psi$
- We write $\langle L, U \rangle \models \varphi$ if $\langle L, U \rangle \models_l \varphi$ and $\langle L, U \rangle \models_l \varphi$
- Under the restriction to modal defaults and modal default theories, se-pairs characterize the entailment relation in S4F

First, the semantics simplifies!

Se-pairs

- $\langle L, U \rangle$ - L, U are propositional theories closed under propositional entailment
- Entailment relations $\models_u$ and $\models_l$ for modal defaults
- $\langle L, U \rangle \models_l \varphi$
  - $\varphi = K\psi$, where ψ is propositional
    $\langle L, U \rangle \models_l \varphi$ if $\psi \in L \cap U$
  - Boolean connectives standard
  - $\varphi = K\psi$, where ψ is a modal default
    $\langle L, U \rangle \models_l \varphi$ if $\langle L, U \rangle \models_l \psi$ and $\langle L, U \rangle \models_u \psi$
- We write $\langle L, U \rangle \models \varphi$ if $\langle L, U \rangle \models_l \varphi$ and $\langle L, U \rangle \models_l \varphi$
- Under the restriction to modal defaults and modal default theories, se-pairs characterize the entailment relation in S4F

First, the semantics simplifies!

Se-pairs

- $\langle L, U \rangle$ - L, U are propositional theories closed under propositional entailment
- Entailment relations $\models_u$ and $\models_l$ for modal defaults
- $\langle L, U \rangle \models_l \varphi$
  - $\varphi = K\psi$, where ψ is propositional
    $\langle L, U \rangle \models_l \varphi$ if $\psi \in L \cap U$
  - Boolean connectives standard
  - $\varphi = K\psi$, where ψ is a modal default
    $\langle L, U \rangle \models_l \varphi$ if $\langle L, U \rangle \models_l \psi$ and $\langle L, U \rangle \models_u \psi$
- We write $\langle L, U \rangle \models \varphi$ if $\langle L, U \rangle \models_l \varphi$ and $\langle L, U \rangle \models_l \varphi$
- Under the restriction to modal defaults and modal default theories, se-pairs characterize the entailment relation in S4F

Further simplifications

Se-interpretations and se-models

- An se-interpretation - an se-pair $\langle L, U \rangle$ such that $L \subseteq U$
- Under the restriction to modal defaults and modal default theories, se-interpretations characterize the entailment relation in S4F
- Se-model of a modal default theory I - an se-interpretation $\langle L, U \rangle$ such that $\langle L, U \rangle \models_l I$ and $\langle L, U \rangle \models_u I$

Properties

Strong equivalence

- Let $I', I'' \subseteq \mathcal{L}_K$ be modal DTs. The following conditions are equivalent:
  - I' and I'' are strongly equivalent ($I' \cup I$ and $I'' \cup I$ have the same S4F-expansions for every modal DT I)
  - I and I' are equivalent in the logic S4F
  - I and I' have the same se-models.

Uniform equivalence

- Modal DTs I', I'' are uniformly equivalent if for every $J \subseteq \mathcal{L}$, $I' \cup KJ$ and $I'' \cup KJ$ have the same S4F-expansions.
- Se-models yield a characterization of uniformly equivalent modal DTs

Properties

Strong equivalence

- Let $I', I'' \subseteq \mathcal{L}_K$ be modal DTs. The following conditions are equivalent:
  - I' and I'' are strongly equivalent ($I' \cup I$ and $I'' \cup I$ have the same S4F-expansions for every modal DT I)
  - I and I' are equivalent in the logic S4F
  - I and I' have the same se-models.

Uniform equivalence

- Modal DTs I', I'' are uniformly equivalent if for every $J \subseteq \mathcal{L}$, $I' \cup KJ$ and $I'' \cup KJ$ have the same S4F-expansions.
- Se-models yield a characterization of uniformly equivalent modal DTs

Properties (2)

Modal rules, modal programs

- Modal rule: $\varphi ::= Kp \mid K\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$
  where p is a propositional atom
- A special class of modal DTs
- A simpler modal logic, SW5, can be used instead of S4F
- Simple se-interpretations: pairs $\langle L, U \rangle$, where L and U are sets of atoms, $L \subseteq U$
- SW5 — an alternative to logic here-and-there
  - logic here-and-there discovered for nonmon reasoning by Pearce 1997
  - underlies disjunctive logic programming with the answer-set semantics (Pearce 1997)
  - forms the basis for general logic programming with the answer-set semantics (Ferraris and Lifschitz 2005)

To sum up

Logic here-and-there

- Is the logic of strong equivalence in general logic programming
- Characterizes uniform equivalence in general logic programming
- Non-mon here-and-there = general LP *(Ferraris and Lifschitz)*

SW5 when restricted to modal programs

- Extends logic here-and-there (and so does all what the other one)
- Connectives “classical” (but modality in the language)

S4F when restricted to modal defaults

- Extends SW5 (modal defaults properly extend modal programs)
- Captures several additional nonmonotonic logics
- Is the logic of strong equivalence in these formalisms
- Characterizes uniform equivalence
- As before, connectives “classical” (but modality in the language)

To sum up

Logic here-and-there

- Is the logic of strong equivalence in general logic programming
- Characterizes uniform equivalence in general logic programming
- Non-mon here-and-there = general LP *(Ferraris and Lifschitz)*

SW5 when restricted to modal programs

- Extends logic here-and-there (and so does all what the other one)
- Connectives “classical” (but modality in the language)

S4F when restricted to modal defaults

- Extends SW5 (modal defaults properly extend modal programs)
- Captures several additional nonmonotonic logics
- Is the logic of strong equivalence in these formalisms
- Characterizes uniform equivalence
- As before, connectives “classical” (but modality in the language)

To sum up

Logic here-and-there

- Is the logic of strong equivalence in general logic programming
- Characterizes uniform equivalence in general logic programming
- Non-mon here-and-there = general LP *(Ferraris and Lifschitz)*

SW5 when restricted to modal programs

- Extends logic here-and-there (and so does all what the other one)
- Connectives “classical” (but modality in the language)

S4F when restricted to modal defaults

- Extends SW5 (modal defaults properly extend modal programs)
- Captures several additional nonmonotonic logics
- Is the logic of strong equivalence in these formalisms
- Characterizes uniform equivalence
- As before, connectives “classical” (but modality in the language)

Algebra and nonmonotonic reasoning

Brief overview

- Fitting's work on logic programming
  - Semantics - fixpoints of operators on lattices and bilattices of interpretations
- Abstract algebraic theory of fixpoints of operators and approximation mappings *(Marek, Denecker, T_, 2000)*
- Algebraic counterparts to models, supported models and stable models, their "partial" versions and approximation semantics: Kripke-Kleene and well-founded
- Provides new semantics (ultimate semantics)
- Provides a unified view of DL and AEL
- Explains common themes in NMR research (cf. algebraic characterizations of stratification and splitting)
- Formalizes the notion of a nonmonotone inductive definition *(Denecker)*

Algebra and nonmonotonic reasoning

Brief overview

- Fitting's work on logic programming
  - Semantics - fixpoints of operators on lattices and bilattices of interpretations
- Abstract algebraic theory of fixpoints of operators and approximation mappings *(Marek, Denecker, T_, 2000)*
- Algebraic counterparts to models, supported models and stable models, their "partial" versions and approximation semantics: Kripke-Kleene and well-founded
- Provides new semantics (ultimate semantics)
- Provides a unified view of DL and AEL
- Explains common themes in NMR research (cf. algebraic characterizations of stratification and splitting)
- Formalizes the notion of a nonmonotone inductive definition *(Denecker)*

What's what or how to abstract?

Logic programming algebraically (Apt, Fitting)

interpretations	$\leftrightarrow$	elements of a complete lattice
program P	$\leftrightarrow$	one-step provability operator T_P
models of P	$\leftrightarrow$	prefixpoints of T_P
supported models of P	$\leftrightarrow$	fixpoints of T_P
stable models of P	$\leftrightarrow$	(certain) fixpoints of T_P

Which fixpoints correspond to stable models?

- 2-input one-step provability mapping Ψ_P (Fitting)
- $\Psi_P(I, I) = T_P(I)$ — an *approximating* mapping to T_P
- Gelfond-Lifschitz operator: $GL_P(I) = \mathit{lfp}(\Psi_P(\cdot, I))$
- Well defined since $\Psi_P(\cdot, I)$ monotone
- Stable models — fixpoints of GL_P

What's what or how to abstract?

Logic programming algebraically (Apt, Fitting)

interpretations	$\leftrightarrow$	elements of a complete lattice
program P	$\leftrightarrow$	one-step provability operator T_P
models of P	$\leftrightarrow$	prefixpoints of T_P
supported models of P	$\leftrightarrow$	fixpoints of T_P
stable models of P	$\leftrightarrow$	(certain) fixpoints of T_P

Which fixpoints correspond to stable models?

- 2-input one-step provability mapping Ψ_P (Fitting)
- $\Psi_P(I, I) = T_P(I)$ — an *approximating* mapping to T_P
- Gelfond-Lifschitz operator: $GL_P(I) = lfp(\Psi_P(\cdot, I))$
- Well defined since $\Psi_P(\cdot, I)$ monotone
- Stable models — fixpoints of GL_P

Approximating mappings

Definition

- L – 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
- If O is an operator on L such that $O(x) = A(x, x)$, then A is an *approximating mapping for* O.

Approximating mappings (2)

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

$$A(x,z) \leq A(z,z) \leq A(y,z) \text{ and } A(z,y) \leq A(z,z) \leq A(z,x).$$

- Consequently

$$A(x,z) \leq O(z) \leq A(y,z) \text{ and } A(z,y) \leq O(z) \leq A(z,x),$$

- That is, pairs $(A(x,z), A(y,z))$ and $(A(z,y), A(z,x))$ *approximate* $O(z)$.

Approximating mappings (3)

Basic properties

- Every operator O has an approximating mapping:

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

- Approximating mappings are not unique (in general)
- If O is monotone, let $C_O(x,y) = O(x)$, for $x,y \in L$
- If O is antimonotone, let $C_O(x,y) = O(y)$, for $x,y \in L$
- In each case, C_O is an approximating mapping for O — *canonical* approximating mapping

Stable operator, stable fixpoints

- 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 $y \in L$:

$$S_A(y) = lfp(A(\cdot, y))$$

- An element $x \in L$ is an *A-stable fixpoint* of O if $x = S_A(x)$
- $St(O, A_O)$ — the set of A-stable fixpoints of O

Back to LP for a moment

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

Only now we do not have a single fixed approximating mapping

Stable operator, stable fixpoints

- 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 $y \in L$:

$$S_A(y) = lfp(A(\cdot, y))$$

- An element $x \in L$ is an *A-stable fixpoint* of O if $x = S_A(x)$
- $St(O, A_O)$ — the set of A-stable fixpoints of O

Back to LP for a moment

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

Only now we do not have a single fixed approximating mapping

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 $T_{P \cup R} = T_P \vee T_R$

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

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 $T_{P \cup R} = T_P \vee T_R$

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:
  - **SE1:** $x \leq y$
  - **SE2:** $P(y) \leq y$
  - **SE3:** $A_P(x, y) \leq x$
- $SE(P, A_P)$ — the set of all se-pairs for P w/r to A_P

Generalize se-models by Turner

- Lattice of interpretations (sets of atoms)
- Operator T_P with an approximating mapping Ψ_P
  - **SE1:** $X \subseteq Y$
  - **SE1:** $T_P(Y) \subseteq Y \quad \rightarrow \quad Y$ is a model of P
  - **SE1:** $\Psi_P(X, Y) \subseteq X \quad \rightarrow \quad X$ is a prefixpoint of $\Psi_P(\cdot, Y) \quad \rightarrow \quad X$ is a model of P^Y

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:
  - **SE1:** $x \leq y$
  - **SE2:** $P(y) \leq y$
  - **SE3:** $A_P(x, y) \leq x$
- $SE(P, A_P)$ — the set of all se-pairs for P w/r to A_P

Generalize se-models by Turner

- Lattice of interpretations (sets of atoms)
- Operator T_P with an approximating mapping Ψ_P
  - **SE1:** $X \subseteq Y$
  - **SE1:** $T_P(Y) \subseteq Y \quad \rightarrow \quad Y$ is a model of P
  - **SE1:** $\Psi_P(X, Y) \subseteq X \quad \rightarrow \quad X$ is a prefixpoint of $\Psi_P(\cdot, Y) \quad \rightarrow \quad X$ is a model of P^Y

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

Converse result

It holds. But a stronger result holds, too!

- 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,
  $St(P \vee R, A_P \vee C_R) = St(Q \vee R, A_Q \vee C_R)$

Uniform equivalence (Eiter, Fink)

Definition (not much choice left, really)

- P and Q are *uniformly equivalent* with respect to (A_P, A_Q), $P \equiv_u Q$ w/r to (A_P, A_Q), if for every *constant* operator R

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

- In the LP setting: extensions by arbitrary sets of facts
- Relevant to query optimization in databases

Characterizing uniform equivalence

Theorem

- P and Q — operators on a complete lattice L
- A_P and A_Q — approximating mappings for P and Q, respectively
- $P \equiv_u Q$ w/r to (A_P, A_Q) if and only if
  - for every $y \in L$, $P(y) \leq y$ if and only if $Q(y) \leq y$
  - for every $x, y \in L$ such that $x < y$ and $(x, y) \in SE(P, A_P)$, there is $u \in L$ such that $x \leq u < y$ and $(u, y) \in SE(Q, A_Q)$
  - for every $x, y \in L$ such that $x < y$ and $(x, y) \in SE(Q, A_Q)$, there is $u \in L$ such that $x \leq u < y$ and $(u, y) \in SE(P, A_P)$

Another characterization

Ue-pairs

- An se-pair $(x, y) \in SE(P, A_P)$ is a *ue-pair* for P with respect to A_P if
  for every $(x', y) \in SE(P, A_P)$ such that $x < x'$, $x' = y$
- $UE(P, A_P)$

Theorem

- L — a complete lattice such that its every subset has a maximal element
- $P \equiv_u Q$ w/r to (A_P, A_Q) iff $UE(P, A_P) = UE(Q, A_Q)$

Another characterization

Ue-pairs

- An se-pair $(x, y) \in SE(P, A_P)$ is a *ue-pair* for P with respect to A_P if
  for every $(x', y) \in SE(P, A_P)$ such that $x < x'$, $x' = y$
- $UE(P, A_P)$

Theorem

- L — a complete lattice such that its every subset has a maximal element
- $P \equiv_u Q$ w/r to (A_P, A_Q) iff $UE(P, A_P) = UE(Q, A_Q)$

Miscellaneous results

- Let P and Q be monotone operators on a complete lattice L. Then $P \equiv_s Q$ w/r to (C_P, C_Q) iff P and Q have the same prefixpoints.
- Let P and Q be monotone operators on a complete lattice L. Then $P \equiv_u Q$ w/r to (C_P, C_Q) iff $P \equiv_s Q$ w/r to (C_P, C_Q).
- Let P and Q be antimonotone operators on a complete lattice L. Then $P \equiv_s Q$ w/r to (C_P, C_Q) iff P and Q have the same prefixpoints and for every prefixpoint y of both P and Q, $P(y) = Q(y)$

Comments and further questions

- Our results generalize results from logic programming
- Also: imply results on equivalence for default logic and a version of autoepistemic logic (with strong expansions of Denecker, Marek and T_)
- The same characterizations as those obtained through logic S4F
- Any direct connection between S4F and approximation theory?
- Is there an algebraic generalization of the logic S4F?

Comments and further questions (2)

- Other classes of extending operators
  - should contain constant operators but not simple operators
  - one possibility (not too many come to mind): antimonotone operators
- Relativized equivalence
  - An operator R on L is a *y-operator* if it is determined by an operator on the complete lattice

$$\{x \in L : x \leq y\}$$

  - By allowing only y-operators as extending operators, we obtain strong and uniform *y-equivalence*
  - These concepts generalize corresponding notions proposed for logic programs by Eiter, Fink and Woltran
  - Work on characterization theorems in progress

Thank you!